NTNU – Trondheim
Norwegian University of
Science and Technology

CFD Analysis of the Flow around the Top Fin of the Flumill Tidal Turbine

Håkon Bartnes Line

Master of Energy and Environmental Engineering
Submission date: June 2013
Supervisor: Helge Ingolf Andersson, EPT

Norwegian University of Science and Technology
Department of Energy and Process Engineering

Sammendrag

OpenFOAM er en programvarepakke for numeriske strømningsberegninger som benytter seg av åpen kildekode, og som er tilgjengelig for gratis nedlasting. Programvaren er i rask utvikling, og er i ferd med å få en vid brukerbase innen både akademia og industrien. I dette prosjektet anvendes OpenFOAM på en industriell problemstilling, nemlig optimalisering av utformingen til toppfinnen på en tidevannsturbin, med henblikk på å minimere strømningsinduserte vibrasjoner.

To profiler, finnens nåværende fasong og et mer strømlinjeformet NACA-0033 profil, gjennomgår såkalte α-sveip, sakte 180-graders rotasjoner i forhold til strømningsretningen, mens kreftene på dem registreres. Utifra denne informasjonen tegnes profilenes hydrodynamiske "signaturer" i tids- og frekvensplanet, og en anbefaling gjøres på bakgrunn av disse.

For å validere simulasjonsoppsettet gjennomføres en rekke simulasjoner for et NACA-0012 profil ved forskjellige innfallsvinkler, samt for en sirkulær sylinder. Alle simulasjonene gjennomføres både som todimensjonale URANS-simulasjoner og som tredimensjonale DES-simulasjoner, og resultatene fra disse sammenlignes. DES-simulasjonene fungerer ikke som forventet og gir upålitelige resultater i visse strømningssituasjoner, men resultatene fra URANS-simulasjonene er i stand til å fylle hullene der pålitelige DES-data mangler.

Abstract

OpenFOAM is an open-source software suite which is available as a free download. The software is under rapid development, and is gaining a wide user base in both industry and academia. In the current project, OpenFOAM is used on an industrial case, namely the optimization of the design of the top fin on a tidal turbine, with the intent of minimizing vortex-induced vibrations.

Two different profiles, the original design and a more streamlined NACA-0033 foil, are taken through so-called α-sweeps, slow 180-degree rotations relative to the flow direction, during which the forces on them are recorded. From this information, the profiles' hydrodynamic "signatures" in the time- and frequency-plane are drawn up, and a recommendation is made based on these.

In order to validate the computational setup, a series of simulations for a NACA-0012 foil at various angles of attack, and for a circular cylinder, are performed. All the simulations are performed as both two-dimensional URANS-simulations and as three-dimensional DES-simulations, and their results are compared. It's found that the DES simulations don't perform as desired, but that the results from the URANS simulations are capable of filling in the blanks wherever the DES-data is found lacking.

Table of Contents

List of Tables

List of Illustrations

1 - Introduction

1.1 – A short primer on tidal power

1.1.1 – Power potential

The tides as a source of renewable energy have the potential to play a big part in greening the energy supplies of many coastal countries. In Europe, the countries bordering on the North Sea basin and the English Channel are the ones with the biggest potential resources available to them. Measurements on tidal currents in the area indicate that an average power of 60 GW enters the North Sea from the north between Malin Head in Ireland and Florø, Norway. From the west, the currents entering between the southwestern tip of Ireland and the northwestern tip of Brittany, France represent a potential of 190 GW [9].

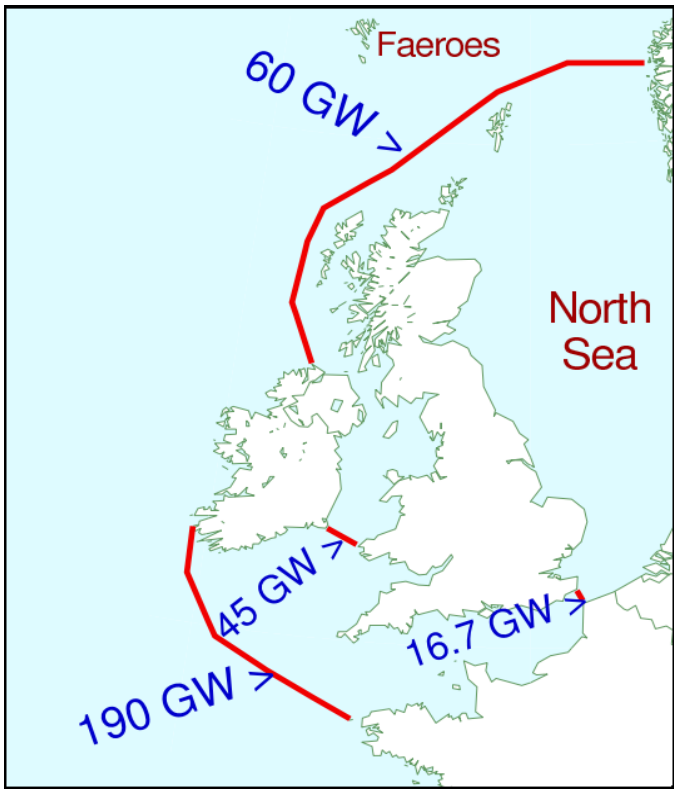

Illustration 1: Tidal power potential of North Sea basin. Taken from [9].

Unlike most other energy sources, the tides don't derive their energy from sunlight. They are instead governed by the motions of the sun and moon relative to the Earth, and the timing and strength of their comings and goings can be therefore be predicted years in advance. Its extreme regularity and predictability is one of its main advantages of this energy source over comparable energy sources like the sun, wind and waves. It lets tidal power be more easily integrated into the national energy grid, and it also makes for less stringent design requirements. Because there are no "freak tides" to have to take into account, tidal power plants, unlike wind or wave turbines, wouldn't have to be engineered for withstanding conditions much more severe than what is found under ordinary operating conditions [9].

1.1.2 - Tidal barrages

Despite these apparent advantages, tidal power remains almost completely unexploited. To date, only two industrial scale tidal power plants exist in the world: One at la Rance on the French side of the English Channel, and the other at Sihwa Lake in South Korea, which was finished in the summer of 2011. Both of these power plants are tidal barrages with about 250 MW of installed power [4, 8]. The barrage at la Rance has been in operation since 1966, so the technology is thoroughly proven.

However, the slow pace of development indicates that they're not competitive compared to other energy technologies. The low head produced by the tidal variations means that the barrages need to draw on

very large tide pools to be able to sustain a high power production in the six hours between tides. The tidal range at la Rance is 11 m, one of the largest in Europe. The area of its tide pool is 22.5km^2, and the plant outputs an average power of 60MW, which means that the power available per unit area of tide pool is 2.7W/m^2 [4, 9]. Tidal barrages are therefore huge infrastructure projects that require large initial investments, and they also have a substantial environmental footprint. Building dams across large estuaries, bays, fjords or wetlands to turn them into reservoirs is controversial, and this might help explain why so few of these projects have been realized.

1.1.3 - Tidal stream farms

Another approach to tidal power extraction which is being explored, though still at the prototype stage, is one that borrows more from wind power than hydro power. It consists of creating tidal stream farms, large arrays of underwater turbines which are anchored to the sea floor and driven by the tidal currents. Though these currents don't reach the same velocities as the wind, they still carry a lot of energy, as water is a thousand times more dense than air. Tidal farms can therefore attain quite high power densities, meaning that large amounts of power can be extracted with relatively little hardware.
Tidal farms don't depend on reservoirs, so this technology scales better than tidal barrages. This means that smaller resources can be exploited with the same efficiency as big ones, and the farms would potentially have less of an impact on the local environment. The tidal turbines would also typically be submerged, and therefore not visible from the surface.

Flow velocity		Kinetic energy flux
[m/s]	[knots]	[W/m^2]
0.5	1	1
1	2	8
2	4	60
3	6	200
4	8	500
5	10	1000

Table 1: Kinetic energy flux of a water current at different velocities.

The kinetic energy flux, measured in W/m^2, increases as the cube of the flow velocity. Some values for different flow speeds are given in Table 1. These are peak values, and the speed of the tidal currents will vary during the day and according to the phase of the moon.

For a better idea of the energy available to a tidal farm, consider a location where the velocity of the tidal current at spring tide is 2.9 knots (1.5 m/s) and that at neap tide is 1.8 knots (1 m/s). The time evolution of the tidal current velocity at this location and the kinetic energy flux that this current carries are plotted in Illustration 2

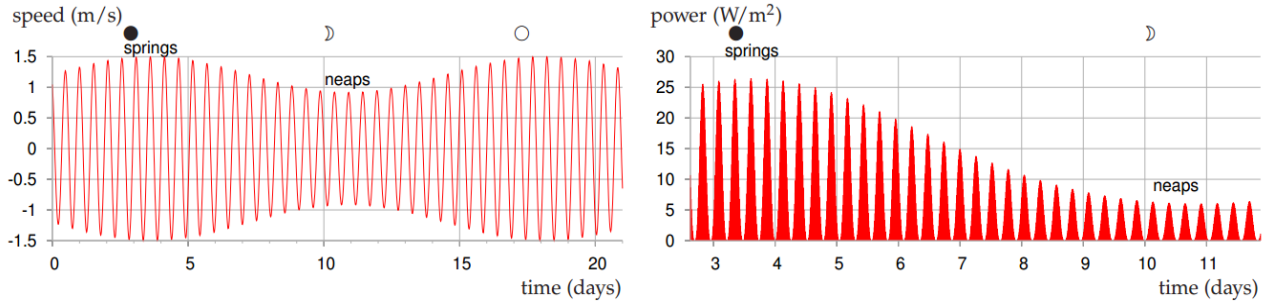

Illustration 2: Variations in the velocity of the tidal current at a typical location plotted over one lunar cycle (left). The kinetic energy flux of this current plotted over a 12-day period (bottom). Both illustrations from [9]

The average power density at this location, which is the area of the graph on the right in figure 2 averaged over a lunar cycle, would be 6.4 W/m^2. This number represents the average amount of power contained in the flow. The actual amount that could be extracted if a location like this were to be developed would be somewhat less. For comparison, Smøla wind farm in Norway has an average yearly output of 356GWh, and covers an area of 18 km^2 [5]. This represents a power density of 2.3 W/m^2 when averaging over the whole area of the wind farm.
These numbers show that, although tidal power is not some "magic bullet", it does have significant potential as a renewable energy source. To date, this potential has largely gone unexploited, but as the technology matures to make its extraction economically viable, tidal power should take a prominent and natural place in the world's renewable energy mix, alongside more conventional sources like wind and hydro. Tidal stream turbines look like the more promising approach in that respect, and it's one of these designs that will be the subject of the present study.

1.2 - The Flumill turbine

One of the experimental design relying on tidal currents is the Flumill tidal turbine currently under development in Arendal, Norway. The main body of the turbine is made up of two counter-rotating helices (4) which are mounted side by side. The axis of each helix is connected to a generator (3) which is in turn mounted on a pivoting steel frame (2) anchored (1) to the sea floor. The pivot allows the helices to adjust their angle with the tides, so that they can extract energy at both ebb and flow.

The helices are connected at the top by a transversely mounted top fin (5). Like the helices, the top fin is buoyant, and it controls the stability and operational angle of the system, which is expected to be between 25 and 50 degrees.

From a technical standpoint, the turbine is quite simple. The helices are made of a composite material, cast in a single piece. The turbine is passively regulated, leaning away from the incoming tide like a weather vane, with the operational angle determined by the balance between the lift, drag and buoyancy forces at the design stream velocity. The only moving parts on the turbine are the hinge in the base and the helices themselves. As the fastest-moving parts of the helices have a velocity about equal to that of the incoming flow, they don't pose any danger to marine life [6].

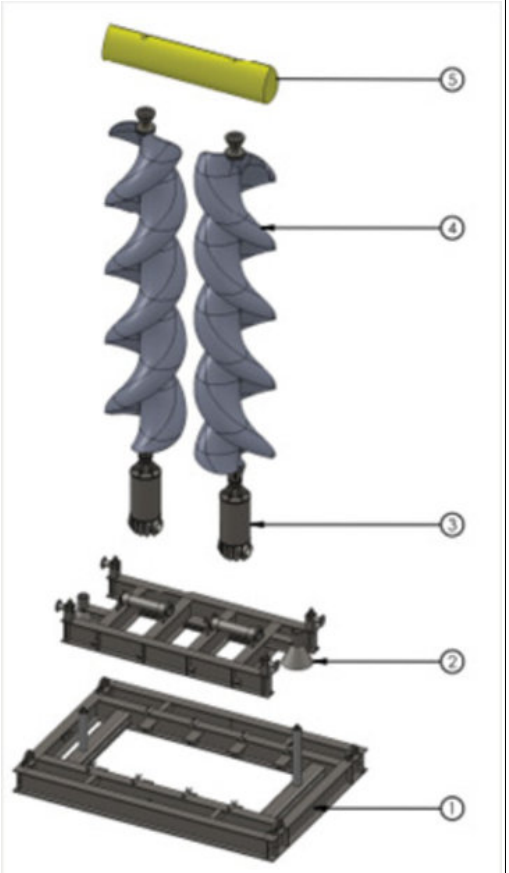

Illustration 3: Design of the Flumill turbine. Taken from [6].

1.3 – Present work

In its current design iteration, the top fin of the turbine is fixed to the turbine axes and has a blunt, vaguely elongated shape. This causes the incoming flow to detach from the fin's surface before reaching the trailing edge, forming vortices that are shed into the wake in a periodic fashion.

The periodic vortex shedding induces vibrations in the turbine structure which may be of a frequency or amplitude sufficient to interfere with the operation of the turbine, or to cause fatigue damage in the long term. Determining the nature of these vortex induced vibrations will therefore be one of the goals of the present work.Ways to reduce the vortex shedding will also be investigated. Since the unsteady wake is caused by the flow detaching from the fin's surface due to strong adverse pressure gradients, the focus will be on reducing these pressure gradients, either through streamlining the fin's shape, reducing its angle of attack to the flow, or a combination of both of these. The fin's angle of attack to the flow could be reduced by replacing the fixed connection between the top fin and helices with a hinge allowing a limited range of rotation θ_t, as shown in Illustration 4.

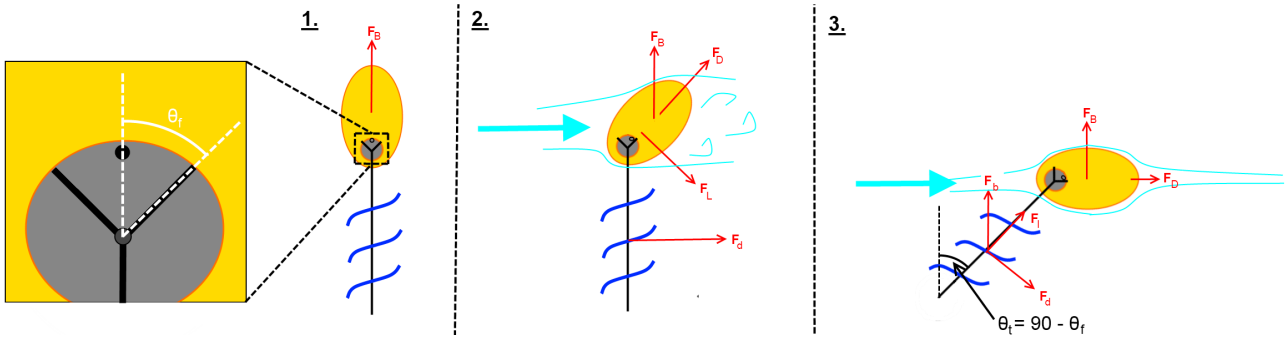

Illustration 4: Rough sketch of a hinge solution for reducing the fin's angle of attack to the flow.

The flow around the top fin will be simulated in OpenFOAM, an open source CFD software package. Two-dimensional unsteady RANS simulations will be performed as a first pass at the problem, and to get an estimate of the frequency and amplitude of the primary vortex shedding instability. Three-dimensional simulations using the Detached Eddy Simulation (DES) procedure developed by Spalart et al. [21] will then be performed on the different fin designs to get a better idea of the forces they would be exposed to at different angles of attack.

2 - Theory

2.1 – The OpenFOAM flow simulation software

OpenFOAM (Field Operation and Manipulation) is a free, open source flow simulation software suite. It was first released in 2004 by OpenCFD ltd., which have been responsible for maintaining and upgrading the code since. They are now a subsidiary of the commercial ESI group. However, the copyright to the source code is held by the non-stock, non-profit OpenFOAM foundation, which was established in 2011 to keep OpenFOAM in the public domain. The copyright is renewed with every new release, and the software is distributed under the GNU public licence, meaning that it's free to use, distribute and modify, but not commercialize. Any third party releases of additions or modifications to the code must therefore be to the public domain, for free use.

OpenFOAM is written as a C++ library, and takes advantage of the object-oriented programming approach offered by that language. It's therefore highly modular, something which lends itself well to the open source format. There are more than a dozen solvers available in the standard release, which can handle most common flow types, from potential flow, to flow involving multiple phases, free surfaces, chemical reactions, solid particles etc. Solvers for electrodynamics, solid mechanics and finance also exist. OpenFOAM also includes a compiler which supports dynamically linked libraries, meaning that custom code can be linked to the solver and included at runtime.

Several third party offerings are in common use among the OpenFOAM community, like swak4Foam and PyFoam, which allow users to add custom functionality without having to write and compile C++-code. HelyxOS is a graphical user interface which is intended to lower the bar for new users, while the OpenFOAM Extend project is a large library which adds extra functionality for advanced users.

Though the wide availability of user-generated modules and other content is a definite plus, open source software does have some downsides, with one of the main problems being the general lack of documentation and user support. Some tutorials and documentation does exist, but the user must often turn to internet forums or look at the source code in order to learn how to use a feature or bypass a problem. There are utilities implemented in OpenFOAM for which there exists absolutely no documentation, and which are the subjects of many dead-end threads in the user forums.

OpenFOAM comes with two grid generators – blockMesh, which creates simple, block-structured grids based on detailed instructions from the user, and snappyHexMesh, an unstructured grid generator which has been used for the current project, and which is described in more detail in Appendix 1.

Flow cases are stored in their own directories, which must contain the following three subdirectories: *"0"*, which contains the initial and boundary conditions, *"constant"*, which contains the grid, physical properties of the flow and certain other files, and finally *"system"*, which contains all files related to the solution procedure itself. A minimum setup for an LES case is shown in Illustration 5, but it can be extended with files for run-time sampling, dynamic grid movement, grid generation and more.

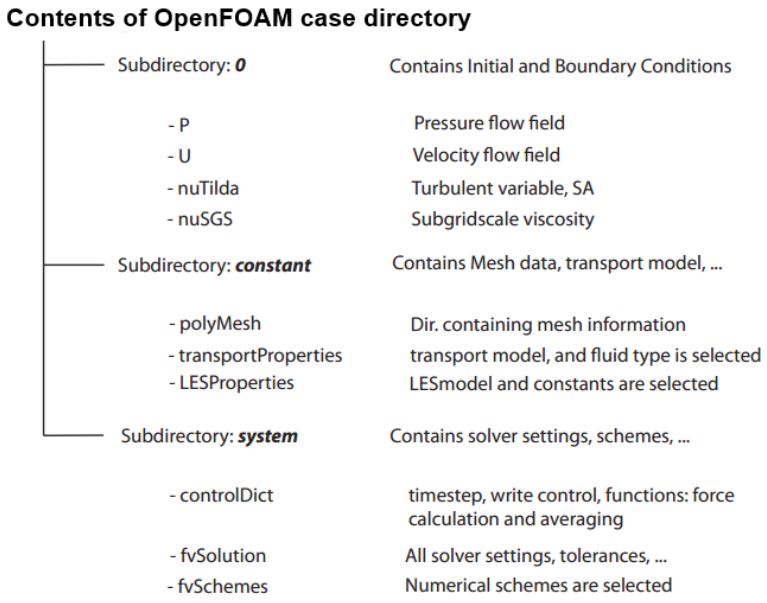

Illustration 5: Minimum contents of the case directory for a LES/DES in OpenFOAM. Adapted from [25]

2.1.1 - Dynamic grid motion in OpenFOAM

OpenFOAM has several procedures for dynamic grid motion implemented, ranging from simple prescribed point motion and point diffusion, which is suitable for limited mesh movement, to topological changes, which can theoretically accomodate any deformation. Topological changes are effected through instructions that are made up of combinations of nine *primitive grid operations:* the addition, removal or (connectivity) modification of a point, face or cell, respectively. Between them, these nine operations can be used to construct or deconstruct any grid, proving their general applicability [7].
Manually issuing these instructions at each time step wouldn't be practical, and so they have been integrated into higher level procedures called *mesh modifiers*. Four of these are implemented in OpenFOAM:

- *Layer/addition removal*, which is commonly used when simulating reciprocal motion, for example in an internal combustion engine. It tells the solver to add/remove a row of cells when they've reached a certain degree of deformation.
- *Attach/detach boundary*, which turns internal faces into boundary faces, effectively detaching one part of the grid from another.
- *Regular oct-tree refinement*, which is used in the refinement algorithm for snappyHexMesh, a grid generator native to OpenFOAM.
- *Sliding interface*, which is used when two grid regions slide along each other. No cells are added or removed in this procedure, but a one-to-one connectivity is kept up through the modification of connectivity and the addition and removal of points and faces.

2.1.2 - Arbitrary Mesh Interface (AMI)

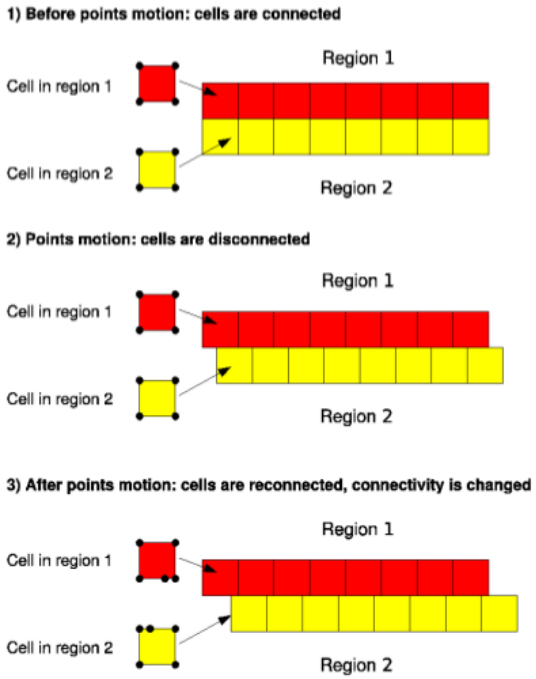

Illustration 6: Recalculation of faces on a sliding mesh interface. Taken from [7]

In the present work, different foil sections will be mounted on a virtual "test bench" and taken through a 180 degree rotation. This requires a grid that's capable of supporting a large range of movement, so topological changes will be necessary. Since the motion is purely rotary, a circular cell zone with a sliding interface will be able to accomodate it.

The name of the sliding interface procedure which is implemented in the regular OpenFOAM releases is the *Arbitrary Mesh Interface (AMI)*. This interface is made by creating baffles, internal boundary faces, in the grid inside the flow field. These baffles can be created either directly by the gridding utility *snappyHexMesh* or indirectly with the *topoSet* and *createBaffles* utilities.

After the baffles have been created, they are then split with the *mergeOrSplitBaffles* utility. This duplicates the faces and points on the baffle, so that they can move relative to each other, as shown in Illustration 6. Finally, the movement of the cell zone inside the AMI is given in a *dynamicMeshDict* dictionary which is put inside the *"constant"* directory of the case file.

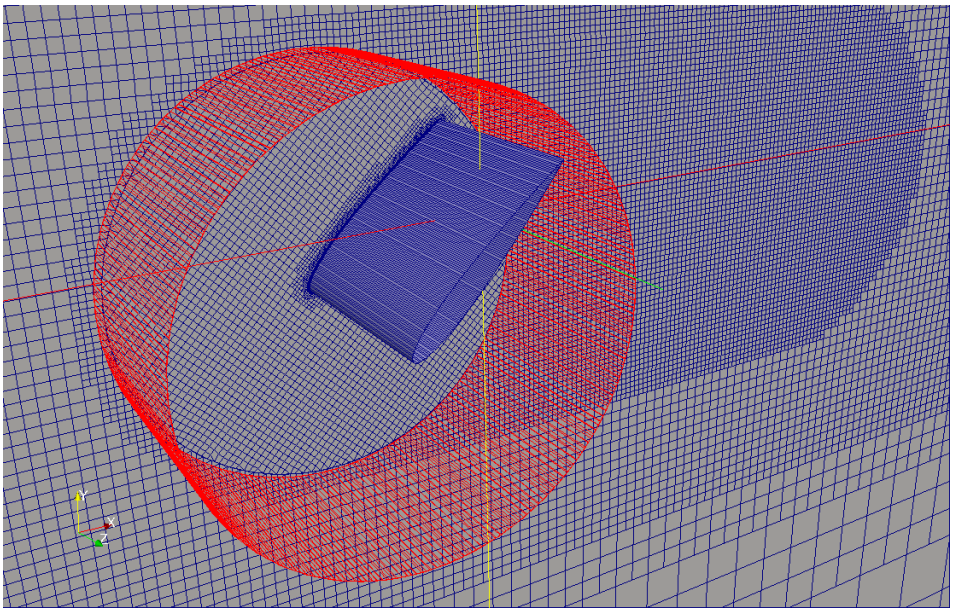

Illustration 7: A grid with a sliding interface generated in snappyHexMesh. AMI highlighted in red.

The case can then be run with the *pimpleDyMFoam* solver, OpenFOAM's solver for single-phase, turbulent flow with mesh motion. At present, motion as a result of forces on the object is not supported for any of the mesh modifiers using topological changes, so the rotational speed must be set beforehand.

2.2 – Maskell's formula for drag correction

The computational domain has a finite extent in the directions normal to the flow, which will cause some confinement effects on the flow field. The boundary conditions will be chosen in order to minimize these effects, but some correction will still be necessary to compensate for the changes in the flow pattern around the object due to the presence of confining walls.

A common method for correcting the drag on bluff bodies in confinement is the one introduced by E.C. Maskell [10]. It's based on the premise that the effect of confinement is to increase the velocity, and hence the dynamic pressure, of the fluid. This lowers the back pressure coefficient on the object, which leads to higher pressure drag. Since its publication fifty years ago, it has become a standard method for calculating wake blockage on bluff bodies, and it has been applied to both experimental and numerical studies [2, 11].

$$\frac{C_D}{C_{D_c}} = \frac{q}{q_c} = 1 + \epsilon C_D \frac{S}{A} \tag{1}$$

These ratios can also be expressed by an empirical relation, which depends on a correction factor ε and the blockage ratio S / A. The factor ε varies between a value of approximately 2.5 for flow around axisymmetric and low aspect-ratio objects, to a little less than unity for flow over two-dimensional objects, for which induced drag is negligible [10]. Values of ε for a two-dimensional flat plate at different angles of attack are given in Illustration 8 below:

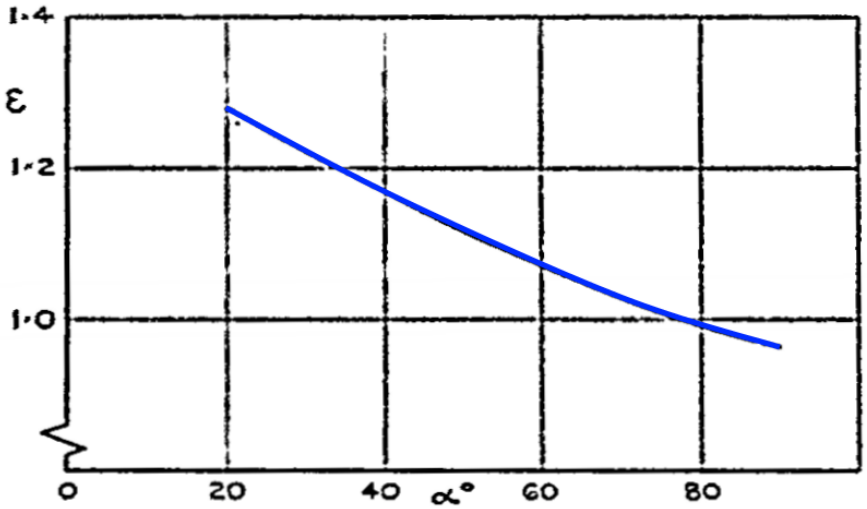

Illustration 8: Correction factor versus angle of attack for flat plate of infinite aspect-ratio. Taken from [10]

2.3 - The Spalart-Allmaras turbulence model

In the following work, the Spalart-Allmaras one-equation turbulence model (SA) will be used throughout to model the diffusion of momentum by turbulence. For the pure URANS simulations, it will be used on all length and time scales over the whole of the computational domain, while for the DES simulations, it will be used in two different capacities, depending on the local properties of the grid and flow field.

In the regions of attached boundary layer flow, where the DES simulations run in RANS mode, the turbulence model will be used in the ordinary way, taking care of turbulent stress modeling at all scales. In the areas of massive flow separation, the simulations will switch to LES mode, and the Spalart-Allmaras model will then only be used as a sub-grid-scale model, limited to modeling the diffusion done by the very smallest turbulent eddies, the ones ones that are too small and of too high frequency to be captured by the computational grid and time step.

Because of the central role played by the SA turbulence model, some pages will be devoted here to a presentation of its contributing terms, constants and the reasoning behind them, before moving on to a presentation of the DES method itself.

2.3.1 – The eddy viscosity assumption

The SA turbulence model is an equation for the transport of turbulent eddy viscosity, so it depends, like most other RANS models, on the Boussinesq assumption. Under this assumption, the Reynolds stress tensor is treated as if it were aligned with the strain tensor of the mean flow. The momentum diffusion by turbulence therefore travels along the same gradients as the diffusion by viscosity, and turbulent activity is modeled as a local increase in fluid viscosity, represented by the eddy viscosity, a spatially (and temporally for unstable flows) varying scalar.

For this reason, the Boussinesq assumption is also known as the eddy viscosity assumption. It's reasonably accurate for attached, near-wall flows and simple two-dimensional turbulent shear layers, where the principal turbulent fluctuations in the layers are aligned well with the mean velocity gradients. The assumption breaks down for more complicated flows like jets and wakes, however, and most eddy viscosity turbulence models tend to greatly overestimate the rate of turbulent mixing for such flows [15].

2.3.2 - Turbulence production and diffusion away from walls

The starting point for the elaboration of Spalart and Allmaras' turbulence model is shown in equation (2) below. It's a simple transport equation containing a material derivative, a production term and a diffusion term. The scalar S which controls the production term represents the local vorticity magnitude of the mean flow field. This quantity was chosen because the model was designed for near-wall flows, where the main mechanism of turbulence production is the vorticity created at the wall due to the no-slip condition.

The possibilities for calibration of this basic model are given by the three constants c_{b1}, c_{b2} and σ, where the subscript b stands for "basic". This naming convention was broken for the constant σ in order to emphasize its role as a Prandtl number. The values for the basic model constants were calibrated in a two-dimensional free-shear flow, and chosen to give the proper velocity profile and shear stresses. A value of $\sigma = 2/3$ was first settled upon as a plausible value for the Prandtl number, yielding the remaining two constant values $c_{b1} = 0.1355$ and $c_{b2} = 0.622$.

$$\frac{D\nu_t}{Dt} = c_{b1} S \nu_t + \frac{1}{\sigma}[\nabla .((\nu + \nu_t)\nabla \nu_t) + c_{b2}(\nabla \nu_t)^2] \tag{2}$$

2.3.3 - Turbulence destruction by inviscid blocking in the log layer

In the vicinity of walls, turbulent fluctuations are damped out by the local pressure field in what is known as inviscid blocking. As the name suggests, this is not related to viscosity and the no-slip condition at the wall, but rather to its non-permeability to flow. This effect makes itself felt in the upper parts of the boundary layer down into the log layer. Below this, in the buffer layer and viscous sublayer, viscous effects from the no-slip condition dominate.

To account for the effect of inviscid blocking and simulate the flow behaviour down to the level of the log layer, Spalart and Allmaras added an eddy viscosity destruction term which depended on wall distance.

According to the law of the wall, the relationship between the eddy viscosity and mean flow vorticity magnitude in the log-layer is as described in equations (3) below [15]. $\kappa = 0.41$ is the von Kármán constant and d_w is the distance to the wall, the only non-local parameter in the model:

$$S = \frac{u_\tau}{\kappa d_w}, \quad \nu_t = u_\tau \kappa d_w \tag{3}$$

This relation suggests a function r, described in equation (4), which equals one in the log layer. This permits a balance between the production and diffusion terms and the destruction term, given that the latter is multiplied by the derived constant $c_{w1} = c_{b1}/\kappa + (1+c_{b2})/\sigma$.

Because of its behaviour outside of the log layer, r wasn't used directly as the function for the eddy viscosity destruction term. It decays too slowly when moving out from the wall and into the freestream, and would give a too high rate of eddy viscosity destruction. It was therefore inserted into the function g, which includes r to the sixth power, and thus decays more rapidly as r tends to zero. The function g, on the other hand, grows very rapidly for values of r greater than one, and therefore had to be bounded from above.

$$r = \frac{\nu_t}{S\,\kappa^2 d_w^2}, \quad g = r + c_{w2}\left(r^6 - r\right), \quad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6} \tag{4}$$

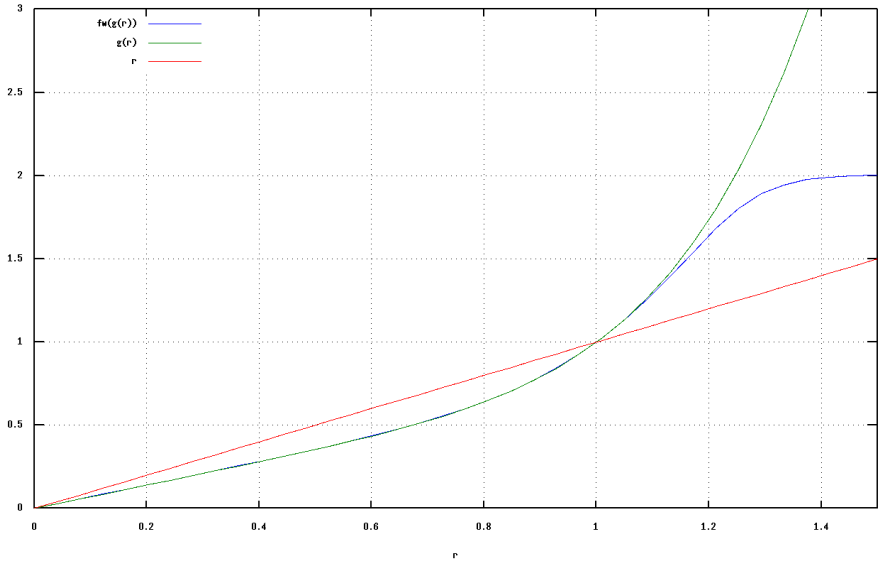

Illustration 9: The near-wall eddy viscosity destruction function plotted against "r".

In order to achieve this smoothly and without clipping, g was inserted into a third function f_w, which stays passive for $r \leq 1$. As r grows past one, f_w flattens out and quickly settles to its upper bound given by the constant c_{w3}, for which 2 was chosen as a reasonable value. Because c_{w1} *was* derived from the other constants in order to balance the terms in the log layer, a third model constant was needed to calibrate the destruction term. The authors chose the skin friction coefficient for the flow over a horizontal flat plate as the calibration criterion, and a value $c_{w2} = 0.3$ was found to give the best result.

2.3.4 - Viscous damping in the lower boundary layer

In the log layer, the eddy viscosity ν_t is a linear function of the wall distance, but this is not the case in the buffer region and viscous sublayer. Here, it falls very rapidly and a very fine grid would be necessary to trace its decay. To avoid the computational expense that this would entail, ν_t was replaced in the model equations by an intermediate variable, defined in such a way that it equals ν_t in the upper parts of the boundary layer, but which also decays linearly all the way to the wall. This new variable $\tilde{\nu}$ is defined by the viscosity ratio χ, which behaves as $\chi = \kappa y^+$ in the region between the log layer and the wall. Since y^+ is proportional to the wall distance and ν is constant for newtonian fluids, $\tilde{\nu}$ also varies linearly with the wall distance.

$$\chi = \frac{\tilde{\nu}}{\nu}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \nu_t = \tilde{\nu}\, f_{v1} \tag{5}$$

The eddy viscosity is related to the new variable by a damping function f_{v1}, where the subscript v stands for viscous. The damping function is only active for low values of χ, for which it provides a proper "law of the wall" distribution for the eddy viscosity. The "reach" of the damping function out from the wall, which determines the log layer intercept, is set by the constant $c_{v1} = 7.1$.

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d_w^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{6}$$

The vorticity magnitude *S* was given the same treatment. Inserting these two new variables into the equations defined above in place of their unmodified counterparts gives a model which is valid in all the different subregions of a turbulent boundary layer. Note that low values of χ are found not only in the inner boundary layer, but also on the outer edges of the turbulent region. The damping functions therefore become active in both places, but since the gradients are less steep outside of the boundary layer, the turbulence model has little impact on the flow field there.

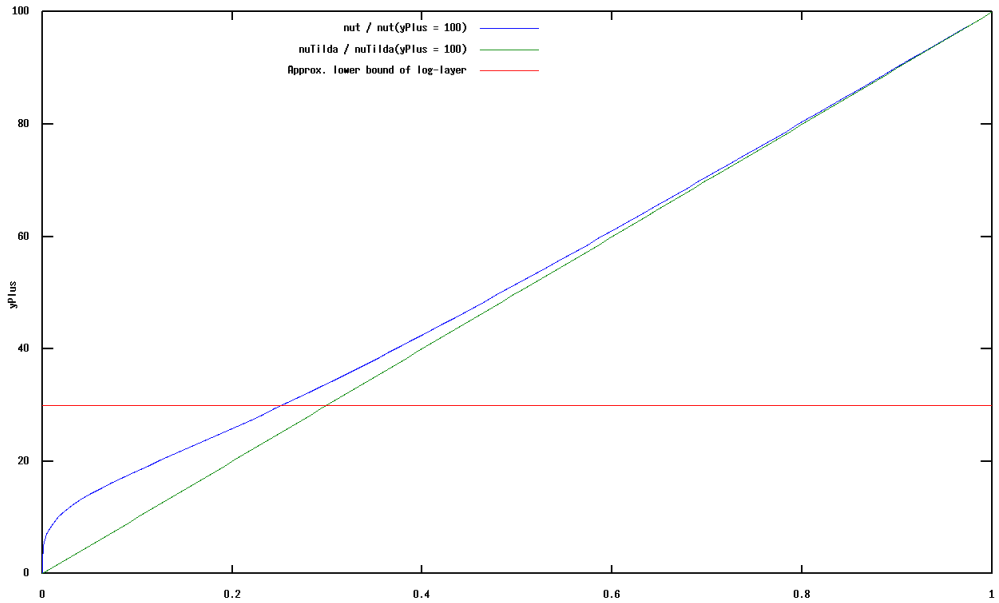

Illustration 10: nut and nuTilda plotted against wall distance, in wall units, and normalized with respect to their values at yPlus = 100.

2.3.5 - The laminar region

The Spalart-Allmaras model is intended to be used with a laminar inflow condition ($\tilde{\nu} = 0$) together with source terms at user-specified locations on the walls that "trip" the boundary layers and lead to transition. This gives a distribution of the skin friction coefficient which is more in line with experimental values, as boundary layers always start out laminar and then transition at some finite distance from the inception point.

This inflow condition can be numerically problematic, because the solution ($\tilde{\nu} = 0$) is an unstable one. The flow field has a tendency to auto-generate eddy viscosity through accumulated numerical errors. This "primes" the flow for transition, causing it to happen sooner than it would in a similar real-life setting. The turbulence model therefore includes the function f_{t2} ,which sets up a small basin of attraction that pulls low values of the eddy viscosity down to zero. It's helped in this by the damping function f_{v1} introduced earlier. The constants c_{t3} and c_{t4} equal 1.2 and 0.5, respectively.

$$f_{t2} = c_{t3} \exp\left(-c_{t4} \chi^2\right) \tag{7}$$

The production term is multiplied by $(1 - f_{t2})$, making it go negative for $f_{t2} > 1$. The edge of the basin of attraction can therefore be found at $\chi = (\log(c_{t3}/c_{t4}))^{0.5} = 0.6039$. Any value of χ lower than this will be pulled down to zero. As mentioned previously, low values of χ can be found not only at the outer edges of the turbulent regions, but also very near the wall. The function f_{t2} is therefore subtracted from the near-wall destruction term as well, in order to avoid upsetting the eddy viscosity balance there.

2.3.6 - Boundary layer transition

The last term to go into the SA model is the trip term, which is responsible for injecting eddy viscosity into the solution and tripping the boundary layers. The trip term is written $\Delta U^2 f_{t1}$, where the subscript t stands for trip. The quantity ΔU^2 is the square of the norm of the difference between the velocity at the trip (which is on a wall) and that at the field point which the equations are being solved for.

$$f_{t1} = c_{t1} g_t \exp\left(-c_{t2} \frac{\omega_t^2}{\Delta U^2} [d_w^2 + g_t^2 d_t^2] \right) \tag{8}$$

The Gaussian in equation (8) gives the trip a roughly semi-elliptical area of effect. The distance from the field point to the trip is given by d_t , and ω_t is the vorticity at the trip. The term g_t in the exponent widens the area of influence of the trip term for coarse grids, making sure that it's active over at least a few grid points in the streamwise direction. It's also multiplied with the exponent, in order to roughly preserve the amount of eddy viscosity accumulated by a particle passing through the trip's area of effect.

$$g_t = min\left(0.1, \frac{\Delta U}{\omega_t \Delta x_t} \right) \tag{9}$$

The constant $c_{t2} = 2$ roughly equals the ratio of average to maximum vorticity $\delta\omega_{wall}/U_{max}$ in a laminar boundary layer. The strength of the trip is proportional to c_{t1} , and a value of $c_{t1} = 1$ was found to reliably trip the boundary layer at a wide range of Reynolds numbers without causing premature transition.

2.3.7 - The complete model and later variants

Putting the above terms together and introducing the modified variables, one gets the full Spalart-Allmaras model, as presented in [24]:

$$\frac{D\tilde{\nu}}{Dt} = c_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\left[\nabla.((\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2\right] - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + f_{t1}\Delta U^2 \tag{10}$$

In the years since the model was first presented, many different versions of it have been implemented, reflecting the needs of the CFD community and the constraints imposed by practical, industrial use.

The trip term, which was presented as a key feature of the model, is rarely used, and has been taken out of many implementations [12]. Instead, it has become common to run the model in fully turbulent mode by having a non-zero eddy viscosity at the inlet, effectively assuming that the boundary layer turns turbulent right from the onset. When used in this way, the f_{t2}-function also becomes unnecessary, and can be taken out.

In fully turbulent mode, phenomena like boundary layer transition, laminar separation and turbulent reattachment aren't considered, and the simulation will likely over-predict the average skin friction coefficient. On the other hand, it relieves the user of the burden of deciding where to trip the flow. Predicting the transition location would be especially difficult for flow over moving objects, where the angle of attack changes with time.

In the original paper by Spalart and Allmaras, it was stated that *"on no account should the model be trusted to predict the transition location"* [24]. However, a new approach was introduced some years later in a paper published by Shur et al. [19], which doesn't require the explicit definition of a trip location. In this tripless approach, the inflow eddy viscosity is still zero, but a small patch of non-zero eddy viscosity is put in the recirculation region behind the object in the initial condition. Most of the eddy viscosity contained in this patch is convected away immediately, but a small amount is retained near the wall, where it diffuses upwards to the separation point. Upon reaching the separation point, it's picked up by the flow and convected out from the wall and back into the recirculation region and the wake, creating a self-sustaining eddy viscosity field. Under certain circumstances, the eddy viscosity also diffuses some distance upstream of the separation point along the boundary layer, causing boundary layer transition ahead of separation.

In a paper co-authored by Spalart, the tripless approach was shown to predict well the separation angle and skin friction coefficient on a cylinder in cross-flow with a turbulent wake [18]. One drawback of this approach is that it only works for separating flows, because a recirculation region is needed in order for some of the introduced eddy viscosity to stay near the body, and not simply be whisked away by the incoming flow. It was because of the dependence on a recirculation region that it was finally decided not to use this method in the present paper, but some simulations were still run on the flow past a circular cylinder, with very good results. The final solution was also found to be insensitive to the amount of eddy viscosity contained in the initial patch, because the amount retained by the body was the same regardless.

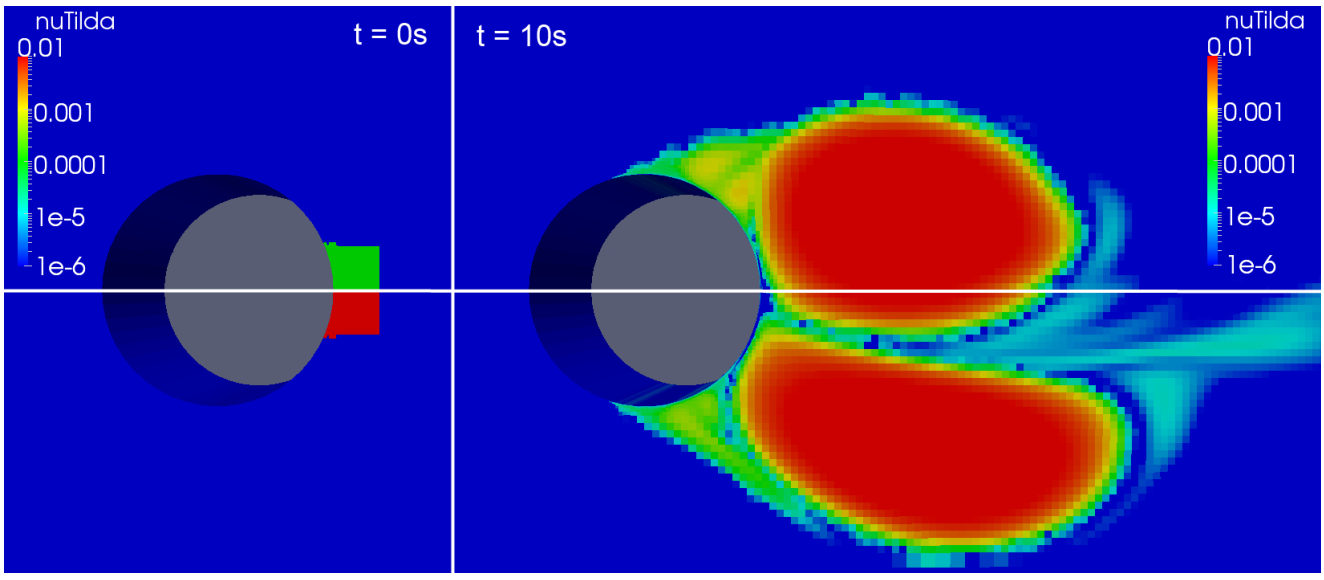

Illustration 11: Comparison of the nuTilda-fields of two tripless simulations, initialized with nuTilda = 1e-4 (top) and nuTilda = 1-e2 (bottom).

2.3.8 - The f_{v3}-implementation

The SA-variant implemented in the OpenFOAM code is the so-called *"fv3"* implementation. It's the same as the standard version, except that it's geared towards fully turbulent mode, and so doesn't include the two trip terms. The viscous correction function for the vorticity magnitude has also been modified:

$$\tilde{S} = f_{v3} S + \frac{\tilde{\nu}}{\kappa^2 d_w^2} f_{v2}, \quad f_{v2} = \frac{1}{(1+\chi/c_{v2})^3}, \quad f_{v3} = \frac{(1+\chi f_{v1})(1-f_{v2})}{\chi}, \quad c_{v2} = 5 \tag{11}$$

This version came about soon after the original model, and was devised to prevent negative values of the production term. As it has been shown to give unusual behaviour at low Reynolds numbers, the NASA turbulence modeling resource does not recommend its use, stating that *"unfortunately, coding of this version still persists"* [12]. For the simulations presented in this paper, the default implementation has therefore been changed back to the original variant (minus the f_{t2}-term).

2.4 - Detached Eddy Simulation

Detached Eddy Simulation (DES) is a hybrid RANS-LES method first introduced by Shur and Spalart in 1997 [21]. The method was defined as follows in a paper from 1999: "*A Detached Eddy Simulation is a three-dimensional unsteady numerical solution using a single turbulence model, which functions as a sub-grid-scale model in regions where the grid density is fine enough for a large-eddy simulation, and as a Reynolds-averaged model in regions where it is not*" [18].

The method's area of application is massively separated flow at high Reynolds numbers. The RANS region, where the grid density isn't fine enough for a large eddy simulation, covers the parts of the flow fields near solid walls where attached boundary layers are expected. The turbulent structures (streaks) in these boundary layers are very small and require an extremely fine grid to resolve. According to one estimate, a bare minimum of 8.000 cells would be necessary to resolve the structures in a cube with sides δ, where δ is the boundary layer thickness [21].

Fortunately, attached boundary layers are just what most RANS models are designed for. They therefore do an excellent job of modelling the turbulent stresses here, based only on information from the mean flow field (and sometimes certain parameters from the flow geometry, like wall distance). Because of this modelling, the grid spacing in the wall-parallel directions, in terms of wall units, can be practically infinite.

As mentioned in section 2.3.1, RANS models based on the eddy viscosity assumption don't perform well for separated flows. A finer grid resolution is therefore put on the regions of the domain where the flow is expected to separate from the body, which lets the DES method switch from RANS to LES mode, and resolve the energy-carrying scales of turbulent motion.

DES is a non-zoned hybrid scheme, meaning that the RANS and LES regions are not tagged beforehand, but are determined by the grid resolution and other parameters. The same turbulence model is used throughout, and so the eddy viscosity field is continuous over the whole domain. It's only its scope which changes as one transitions from the RANS to the LES regions.

2.4.1 - DES97

The first iteration of the method, denoted DES97 from here on, used a single criterion to flip the RANS/LES switch. It was originally implemented with the Spalart-Allmaras model as a RANS/SGS-model, but any turbulence model that includes a characteristic length scale will work [21].

The only modification that was made to the Spalart-Allmaras model in order to implement it in DES was to replace the wall distance from the original formulation with a new length scale. This new length scale was defined as:

$$l_{DES97} = min(d_w, C_{DES} * \Delta), \quad \Delta = max(\Delta x, \Delta y, \Delta z), \quad C_{DES} = 0.65 \tag{12}$$

The parameter Δ represents the maximum cell spacing in the cell where the model is being evaluated, and C_{DES} is a calibration constant. Wherever the distance from a cell centre to the wall is less than C_{DES}*Δ, the length scale is set equal to the wall distance, and the original formulation of the SA model is retrieved. This is a signal to the solver to operate in RANS mode, using the mean flow quantities.

If the distance to the wall is greater than C_{DES}*Δ, the flow solver switches to LES mode. Instantaneous quantities are used, and the SA model functions as a sub-grid scale model with its characteristic length scale proportional to the maximum cell spacing.

2.4.2 - MSD and Grid-induced separation

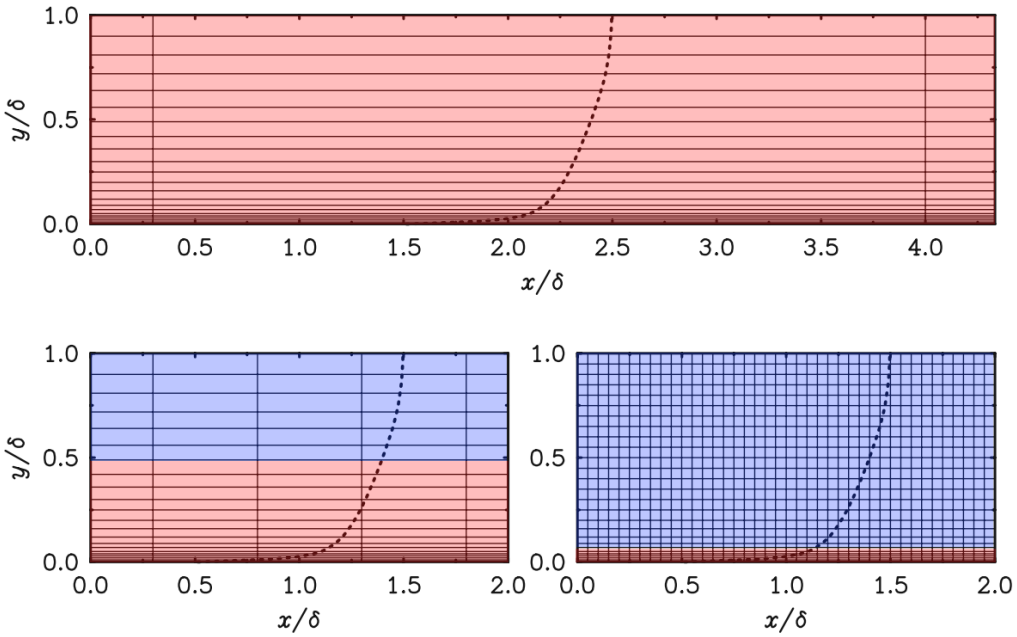

Illustration 12: RANS (red) and LES (blue) regions on three different boundary layers grids. Clockwise from the top: a correctly designed DES grid, a WMLES grid and an ambiguous grid.

For mesh cells in the boundary layer, the cell spacing in the wall-parallel direction $\Delta_{\parallel}$, should typically be at least twice the boundary layer thickness δ. This will insure that C_{DES}* Δ > d_w, and that the DES model stays in RANS mode throughout the boundary layer. Unfortunately, it's sometimes difficult to fulfill this requirement everywhere on the geometry [22].

For low-Reynolds number flow, the boundary layers can grow to be quite wide. If the geometry is also big and complex, creating a grid that meets this requirement on every surface where one expects an attached boundary layer can be difficult and time-consuming. For objects that move relative to the flow, grids that were once perfectly fine can suddenly become ambiguous as boundary layers attach, detach, grow and shrink. Factor in a grid refinement study, and the gridding process becomes almost impossibly constrained, possibly with some very awkward grids as a result.

Ambiguous grids will cause a switch from RANS to LES mode in the middle of a boundary layer. The cells that then find themselves on the wrong side of this interface will typically be too coarse to properly resolve the turbulent stresses that were modeled in the neighbouring RANS cells. This leads to an abrupt, unphysical drop in turbulent stresses called Modeled Stress Depletion (MSD), and it was in fact anticipated when DES was introduced [18]. MSD will cause an underestimation of the skin friction in the affected areas, and may in severe cases cause the flow to separate prematurely, in what is know as Grid-induced Separation (GIS).

2.4.3 - DDES

Spalart et al. addressed this issue with a new version of the DES model, published in 2006 [22]. Delayed Detached Eddy Simulation (DDES) introduced a slight modification to the parameter r from the original S-A model (2.3):

$$r_d = \frac{\tilde{\nu} + \nu}{\tilde{S}\,\kappa^2 d_w^2} \tag{13}$$

The subscript d in this modified parameter represents “delayed”. It's approximately one inside the logarithmic part of the boundary layer, and gradually drops to zero as one moves out through the boundary layer and into the freestream. The kinematic viscosity ν was added to keep the parameter from dropping to zero again on the wall. The parameter r_d is placed inside a delaying function f_d:

$$f_d = 1 - \tanh\left((8 r_d)^3\right) \tag{14}$$

This function uses r_d to the third power as the argument in a hyperbolic tangent. This gives a rapid transition when moving out from the log layer, and keeps the function bounded between zero and one. It's introduced into the modified length scale as follows:

$$l_{DDES} = d_w - f_d * max\left(0, d_w - C_{DES} * \Delta\right) \tag{15}$$

Inside the log layer, where the delaying function is zero, the modified length scale simply becomes the wall distance, as in the unmodified Spalart-Allmaras model, and the solution is locked in RANS mode. Outside the log layer, the original DES length scale is quickly retrieved.

Although these are relatively minor modifications to the model equations, they do represent an important departure from the original DES formulation. Because the value of f_d depends on the turbulent viscosity field and the mean flow velocity gradients, the RANS/LES boundary has now become solution-dependent. The presence of turbulent viscosity becomes self-perpetuating, so that the model will resist switching out of RANS mode. This is of course exactly what the delaying function is supposed to do, but it does introduce the possibility of a branching, with the flow field converging to two different solutions depending on the amount of turbulent viscosity or LES content present in the initial and inflow conditions.

This requires some extra attention on the part of the user, but it's rarely seriously deceiving, as the two possible solutions are radically different. A quick glance at the visualized flow field will therefore make it clear if it has converged to a RANS-heavy state. Many flow solvers with DES functionality also include some form of indicator function to let the user know which mode is active in which grid cells.

2.4.4 - The log-layer mismatch, excessive dissipation of SGS turbulence

The boundary layer grid on the bottom right of Illustration 12 could also be said to be ambiguous, but in this case the ambiguity wouldn't cause MSD even with the original DES formulation, as the grid would be fine enough to resolve the turbulent stresses once the RANS model deactivated. Although the resolution is excessive for a true DES simulation, it represents another use of DES which is gaining ground, namely Wall-Modeled LES (WMLES).

In a DES-WMLES simulation, the RANS region doesn't cover the entire boundary layer, but only the viscous sublayer and the lower part of the log layer. The rest of the flow, beginning from the upper log layer, is resolved by the LES model. This use of DES was pioneered by Nikitin et al. [13], who also discovered the principal weakness of the method, namely that the flow quantities in the modeled and resolved parts of the log layer didn't match at the interface. This log layer mismatch would lead to an under-prediction of the skin friction by as much as 15-20%.

One of the main reasons for the mismatch was found to be the sub-grid length scale used in DES, or rather the fact that it was locked to a single definition across the entire flow domain. Because the decay rates and energy spectra of wall-bounded turbulence are very different to those of the homogeneous, isotropic turbulence found in the freestream, SGS model constants that are calibrated for one region won't work so well in the other.

Explicitly filtered large-eddy simulations can compensate for this by widening the gap between the grid spacing and the energy-containing scales in the freestream, but for an implicitly filtered LES there is no such gap. Furthermore, for DES and WMLES, the cells near the wall, often even the ones in the LES region, are highly anisotropic. Since DES and DDES used the maximum grid spacing instead of the cube root of the cell volume to define the sub-grid length scale Δ, these methods were particularly affected, and had a tendency to cut off many of the energy-containing eddies near the wall.

Finally, an issue was also discovered with the use of Spalart-Allmaras as an SGS model at low Reynolds numbers, because of its damping terms. The sub-grid scale viscosity ν_{SGS} represents the momentum diffusion by turbulent motions which are too small to be captured by the grid. Increasing the grid resolution for a given Reynolds number will increase the resolved stresses at the expense of the modeled stresses, hence causing the sub-grid scale viscosity to decrease. If the length scale becomes too small, the near-wall destruction term will kick in, and quickly pull the sub-grid scale viscosity down to zero. The net effect is an exaggerated rate of turbulence dissipation in these LES regions [22].

2.4.5 - IDDES

In order to deal with these issues, Shur et al. published a second reworking of the DES method, called Improved Delayed DES in 2008 [17]. IDDES resolved the log layer mismatch, and improved support for WMLES by introducing a new length scale and functions to facilitate the switch from RANS to LES mid-boundary layer. The details of the implementation of the WMLES branch are quite involved, and won't be entered into here. The new branch was designed specifically not to interfere with the already established DDES function, so unless significant LES content is introduced in the initial condition or at the inlet, it stays inactive, and the method behaves like a regular DDES.

The problem of uniting wall-bounded and free turbulence in a single implicitly filtered LES was solved by introducing a new sub-grid length scale definition which, in addition to being a function of the local grid spacing, also depended on wall distance:

$$\Delta = min\left(max\left[C_w d_w , C_w h_{max} , h_{wn} \right], h_{max} \right) \tag{16}$$

Here, $C_w = 0.15$ is an empirical constant, h_{wn} is the local grid-cell spacing in the wall-normal direction, and h_{max} is the sub-grid length scale for $d_w \to \infty$. In the original paper, this was set equal to the maximum local grid cell spacing, but since the computational grid should ideally be isotropic away from walls, it doesn't matter much if h_{max} is defined in this way or as the cube root of the cell volume.

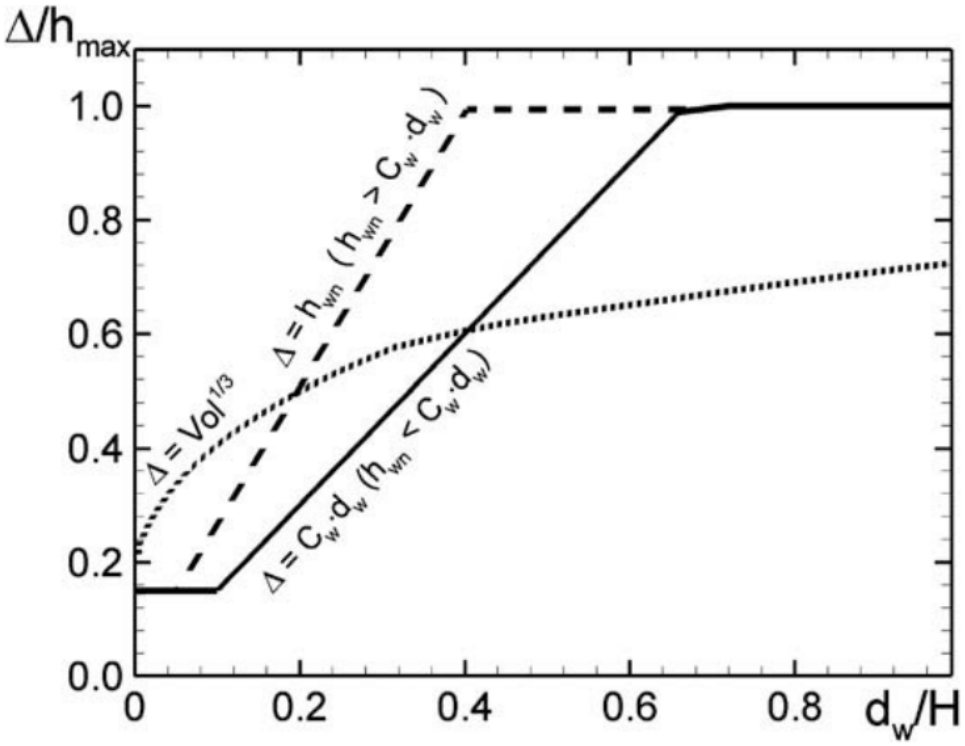

Illustration 13: The two possible evolutions of the IDDES sub-grid length scale, compared to the cube root definition. Wall distance normalized with respect to an arbitrary point in the freestream. Taken from [17]

The above definition lets the length scale Δ vary with wall distance as shown in Illustration 13 above. Very near the wall, $\Delta = C_w * h_{max}$, and it could take two different paths to its "freestream" value, depending on the geometric growth rate of the wall-normal grid spacing of the near-wall cells. If the growth rate $k < (1 + C_w)$, then Δ would follow the solid line, and grow as $\Delta = C_w * d_w$ .If $k > (1 + C_w)$, then Δ would grow at that higher rate, namely as $\Delta = h_{wn} = y_0 k^n$, with y_0 the wall-normal spacing of the first cell at the wall, and n the number of cells out from the wall. The growth rate k shouldn't in any case be higher than about 1.2-1.3.

Spalart et al. claimed another benefit of the new definition of Δ, namely that it would give a steep variation of the sub-grid scale eddy viscosity moving out from the wall. This would have a destabilizing effect on the flow, and quickly generate turbulent content to replace the modeled stresses which were lost on leaving the RANS region.

A second modification was made to the sub-grid length scale involving a low-Reynolds number correction term Ψ, which was introduced in order to keep the destruction terms from kicking in when the LES length scale became too small. All the constant and function definitions are the same as the ones used in the Spalart-Allmaras turbulence model, except for $f_w^* = 0.424$.

$$\Psi^2 = min\left[10^2, \frac{1 - \frac{c_{b1}}{c_{w1}\kappa^2 f_w}\left[f_{t2} + (1 - f_{t2}) f_{v2}\right]}{f_{v1} max\left[10^{-10}, 1 - f_{t2}\right]}\right] \tag{17}$$

If the Spalart-Allmaras model is implemented without the f_{t2}-function, the terms involving it can simply be taken out of the definition of Ψ. *Illustration 14* shows that Ψ equals one for ν_{SGS} greater than about 10ν, but rises quickly for lower viscosity ratios, and reaches its upper bound of *10* for $\nu_{SGS} \approx 0.5\nu$.

$$l_{IDDES} = d_w - f_d * max(0, d_w - \Psi * C_{DES} * \Delta) \tag{18}$$

As equation (18) shows, the correction is applied by simply multiplying it with the LES length scale, effectively making the length scale up to ten times larger in cells where the sub-grid scale eddy viscosity would otherwise have been unable to sustain itself.

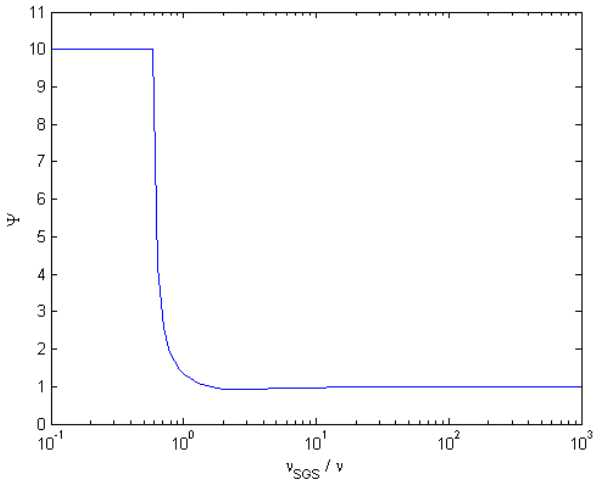

Illustration 14: The low-Re correction term plotted against the viscosity ratio.

2.4.6 – Comments

Apart from the log-layer mismatch, the issues that were addressed by the introduction of the improved DES variants weren't errors in the strictest sense, but the new additions improved the robustness of the method. However, each new version also added another layer of complexity to what was initially a very simple and elegant idea.

Especially the step from DDES to IDDES introduced a lot of complexity, with many empirical functions and constants that give the latter method a very *ad hoc* feel. This impression is strengthened by the fact that IDDES was published only two years after DDES, while nine years separated DDES and DES97. Shur et al. mentioned in their article that an important reason behind their early publication was simply to commit the current state of the art of DES to record, and that they hoped that it would be *"a useful step for better proposals to arise from the community"*[17].

2.5 - DES grids

DES integrates regions with widely different gridding requirements in a single solution, yet unlike some other hybrid methods, it's not zonal in the sense that grid cells are explicitly tagged "RANS" or "LES" beforehand. In the absence of explicit tagging, it's therefore mainly up to the grid to communicate to the solver which branch to follow in any given part of the flow field. This can make designing grids for DES quite challenging, as many different needs have to be met and weighed against each other [23]. The delaying function which comes with later versions of DES brings the solution into play when deciding between RANS and LES mode, and this takes some of the load off grid design.A common way to approach the gridding process is to roughly divide the flow domain into different regions depending on the type of flow that's expected there, and then grid them according to their respective requirements, making sure in the end that there's a smooth transition between each of these regions. The number of subdivision may vary, but for external flows, it's common to distinguish among four main ones – the Euler, RANS, focus and departure regions.

2.5.1 - Euler region

The Euler region usually takes up most of the flow volume, and it has the least stringent gridding requirements. Cells in this region are never supposed to be touched by vorticity or turbulence, and velocity gradients here are minimal. The cells are fairly isotropic, and the flow is nominally LES, though with negligible LES content. The data from the Euler region is not of interest to the solution, and its purpose is to allow the near-body flow values to relax back into the freestream values with minimal interference from the outer boundaries.

2.5.2 - RANS region

The RANS region extends a short distance out from the body in areas where attached boundary layers are expected. It only occupies a tiny portion of the flow volume, usually a fraction of one percent, but it plays a central part in determining the skin friction, pressure distribution and possible boundary layer separation, and as such has a big impact on the solution.

The full range of turbulent stresses are modeled in this region, and normal RANS gridding practices apply. The Reynolds averaging permits an anisotropy of the cells which makes grid refinement relatively cheap, so it's recommended to resolve the boundary layer all the way to the wall for the sake of accuracy. In the wall-normal direction, a first cell spacing of $\Delta y^+ \approx 2$ with a growth rate of 1.2 – 1.3 should be sufficient. In the wall-parallel directions, the cell spacing in wall units is practically unlimited, as long as it can accurately capture the shape of the body and the mean flow gradients.

A relatively coarse wall-parallel grid spacing is actually recommended, as a strong anisotropy clearly

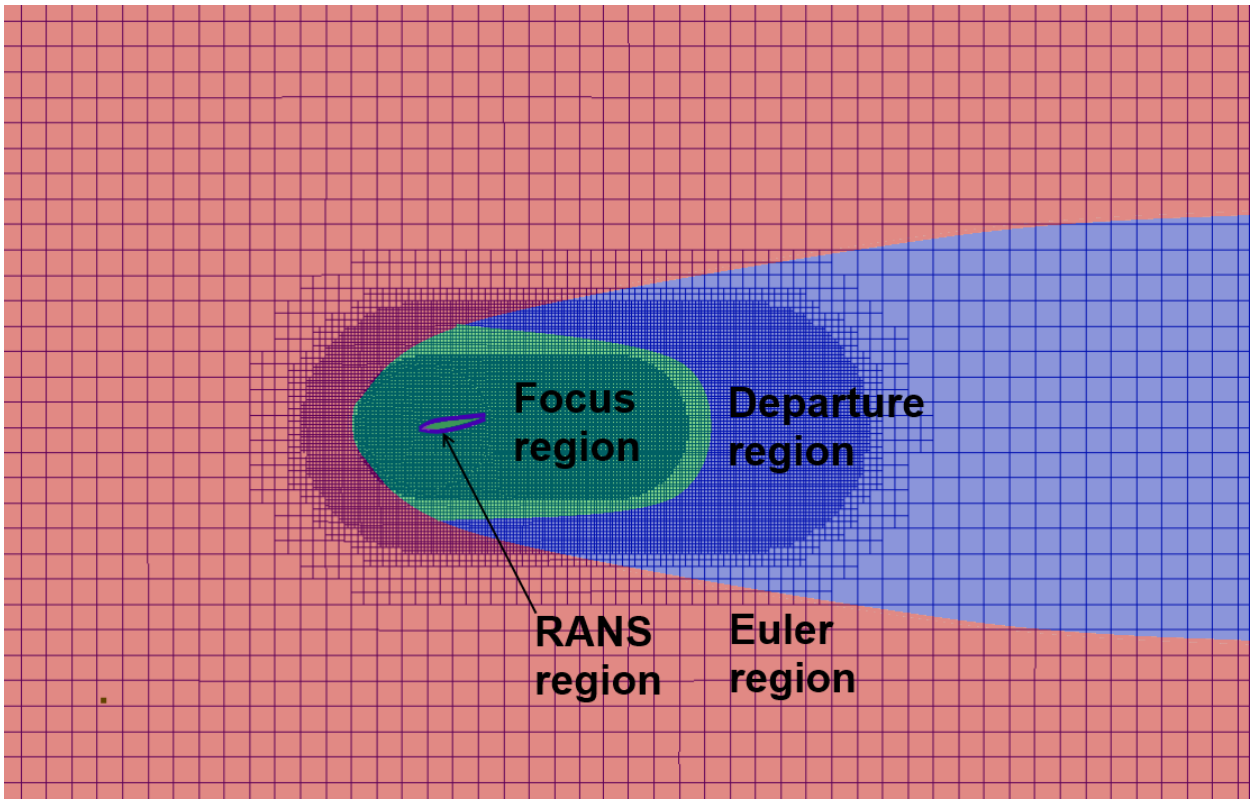

Illustration 15: Rough sketch of the different regions in a DES grid.

marks the cells as belonging to the RANS region. As long as the boundary layer has been satisfactorily resolved, the RANS region can be mostly left out of a grid refinement study [23]. This avoids problems with ambiguous grids, which can be an issue even for DDES if the flow conditions change during the course of the simulation run.

2.5.3 - Focus region

The focus regions is the first of the two regions that have significant LES content. It begins right outside the RANS region and extends downstream into the near-wake. The transition between this region and the RANS region, the so-called “grey area”, is a critical one in any DES study, because the modeled turbulent stresses that fall away when the flow crosses the interface need to be quickly replaced by resolved stresses from the LES. This is mainly a modeling concern – one example of this is the variable sub-grid length scale definition in IDDES, which is supposed to provoke flow instability at the transition. The responsibility of the grid designer is therefore only to make sure that the grid is reasonably smooth here, i.e. that the exact location of the interface can't be predicted beforehand simply by looking at the grid cells.

The resolution requirement in the focus region is the same as for an ordinary implicitly filtered LES. The grid should therefore be fine enough to resolve the energy cascade down to a level where the turbulent eddies no longer impact the bulk flow field. There are no hard rules dictating the cell spacing here, so the proper resolution needs to be found through trial and error. This region should therefore be the focus of any grid refinement study.

Since the smallest length scale that can be meaningfully resolved depends on the maximal dimension of a grid cell, perfectly isotropic (usually cubic) cells give the highest definition at the lowest cost. The focus region has a very high cell density, and it is, particularly for unstructured grids, by far the single biggest contributor to the total cell count. It's therefore desirable to make it as small as one can safely get away with. If one is mainly interested in the forces on a single body, a rule of thumb is that the focus region should extend some distance downstream of the recirculation region, so that no fluid particle that has ever left the focus region can return to a place very near the body.

2.5.4 - Departure region

Further downstream, the focus region gives way to the departure region, which follows the wake all the way to the outlet. The spacing can be a lot coarser here, as we're usually not interested in the exact flow behaviour in this region. The coarsening should be gradual, so that the errors here don't propagate upstream into the focus region. The sub-grid scale eddy viscosity grows very fast as the grid coarsens, further smearing out the gradients, and the flow in the far wake therefore usually takes on the qualities of a quasi-steady RANS. The departure region can be thought of as a kind of calculated downstream boundary condition, and it can in some cases be left out altogether, if the boundary condition on the outlet is well chosen.

3 - Method

3.1 - Computational domain

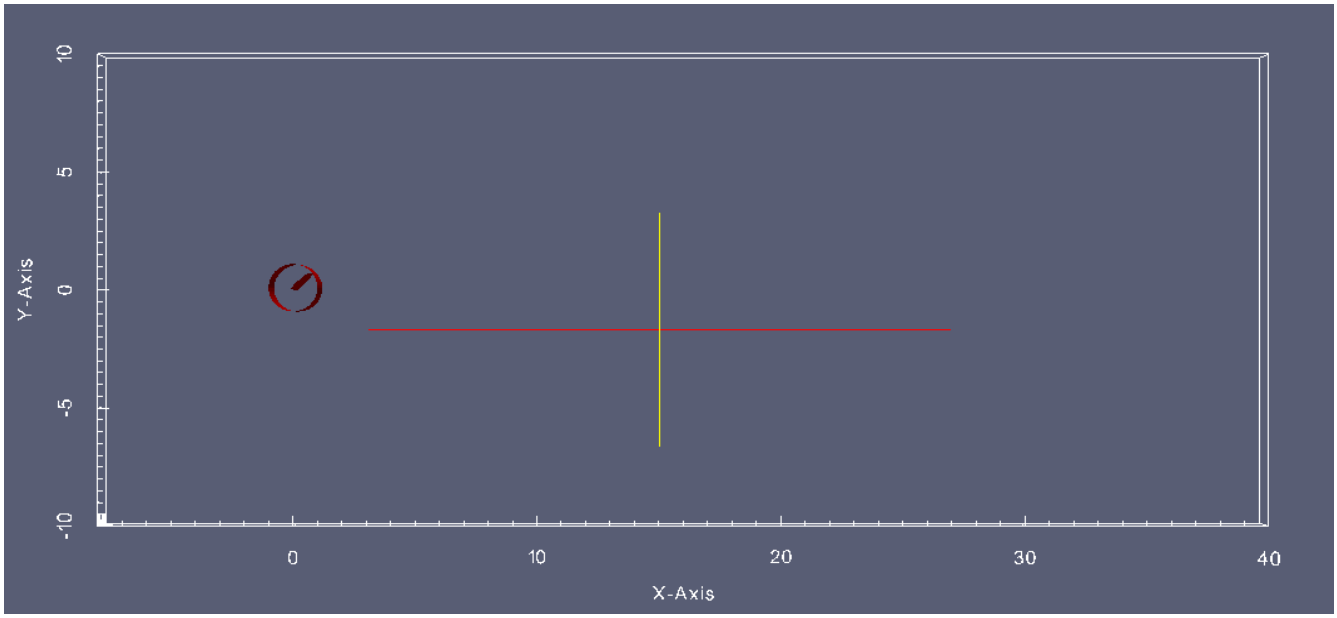

Illustration 16: Side view of the computational domain.

All points and distances in the following description are given in chord lengths. The computational domain represents a foil section of infinite span mounted on a rotating turntable inside a rectangular tunnel. The flow enters in the positive x-direction through the inlet plane at $x = -8$, and exits through the outlet plane at $x = 40$. The crosswise extent of the domain is twenty chord lengths, and the spanwise is one chord length.

The rotating cell zone is a cylinder with a radius of one chord length. It's centered on the origin of the xy-plane, and spans the whole of the domain in the z-direction. It's connected to the rest of the grid through an AMI. The points on the AMI are treated as internal points, so the interface isn't "seen" by the flow field.

A fixed rate of rotation is imposed on the cell zone, causing the foil section contained inside it to change its angle of attack to the incoming flow with time in what is known as an "alpha sweep".

The range of the alpha sweep is 180°, going from -90° to +90° to the flow direction. The cell zone's axis of rotation passes through the chord line of the foil at a distance of one tenth of a chord length inwards from the foil's leading edge.

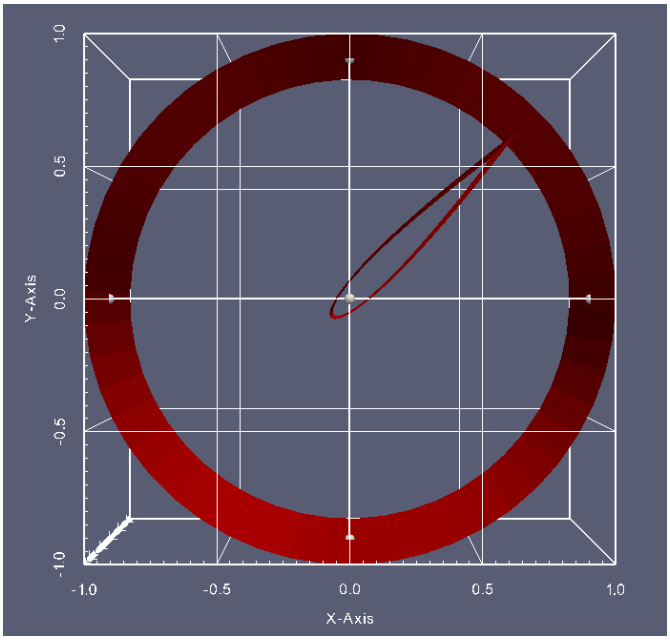

Illustration 17: Close-up of the rotating cell zone.

3.2 - Foil sections

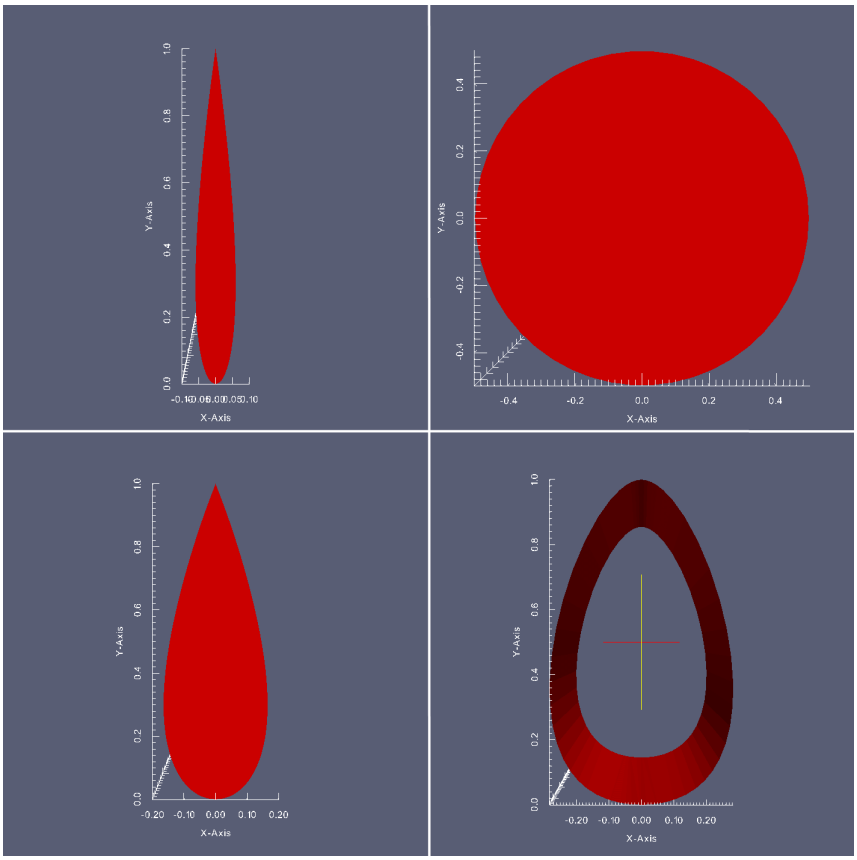

Illustration 18: The different profiles used in the study. Clockwise from top left: NACA-0012, circular cylinder, original top fin design and NACA-0033.

The NACA-0012 profile has been extensively studied in both experiments and numerical flow simulations, and data is available for the whole range of angles of attack from 0° to 90° [20]. It will therefore be used as a test case to validate the solution over an airfoil-shaped object that enters and exits stall. The flow over a circular cylinder will also be used as a validation case to test the simulation's ability to predict the forces on a blunt object where the separation point isn't given by the geometry. The original design of the top fin will be put through an alpha-sweep in order to record the forces on it at different angles of attack. Finally, a sweep will also be performed on the current top fin design and a NACA-0033 profile in order to compare their performance.

The NACA-profiles were generated with the NACA2STL octave script written by Håkon Strandenes [14]. A convenient feature of this script was that it put caps on the ends of the profiles so that they became manifold (closed) surfaces. This made it possible to generate feature-edge meshes from them, which improved feature snapping for sharp edges in snappyHexMesh, OpenFOAM's native grid generator. The surface making up the original top fin design wasn't manifold, but since neither this one nor the cylinder had any sharp edges like the NACA-foils' trailing edges, a surface-edge mesh wasn't necessary for these profiles.

3.3 – Computational grids and time step

The grids were generated in snappyHexMesh, and were therefore unstructured with mainly hexahedral cells. They came in three different degrees of fineness, attained by varying the resolution of the base mesh. The grids were designed according to the principles for DES grids presented in section 2.5. For stability and accuracy, the time step was set to give a maximum CFL number well below one. The fine cells on the trailing edge of the NACA-foils were probably why these grids required a shorter time step than the blunt-body grids.

As the boundary layer (BL) grid added by snappyHexMesh was O-shaped, the sharp trailing edge of the NACA profiles caused the cells there to be skewed, leading the BL grid to "roll up", or unravel, at the trailing edge if the skewness passed a certain threshold. This became a constraint of the gridding process, and the BL grid was optimized to give the lowest possible wall-normal grid spacing Δy at the wall, while giving a smooth transition to the background grid and avoiding "roll-up" at the trailing edge. In order to meet these constraints, the grids for the NACA foils had to be refined an extra level at the wall. More information about BL grid roll-up, and the use of SHM as a DES grid generator can be found in appendix 1.

Name	Cell count	Time step ($\Delta t*L/U_{inlet}$)	Refinement lvl near wall	Δy at wall	Growth rate	No. layers	Δ_{focus}
NACA12Coarse	536k	2.00E-03	6	2.40E-04	1.3	8	0.04167
NACA12Medium	860k	1.25E-03	6	1.80E-04	1.3	8	0.03125
NACA12Fine	1.18M	1.00E-03	6	1.44E-04	1.3	8	0.02500
NACA33	794k	1.25E-03	6	7.10E-05	1.3	10	0.03125
CylinderMedium	637k	2.50E-03	5	2.00E-04	1.3	14	0.03125
CylinderFine	939k	2.50E-03	5	1.60E-04	1.3	14	0.02500
TopFin	612k	1.25E-03	5	2.00E-04	1.3	14	0.03125

Table 2: Overview of the grids used

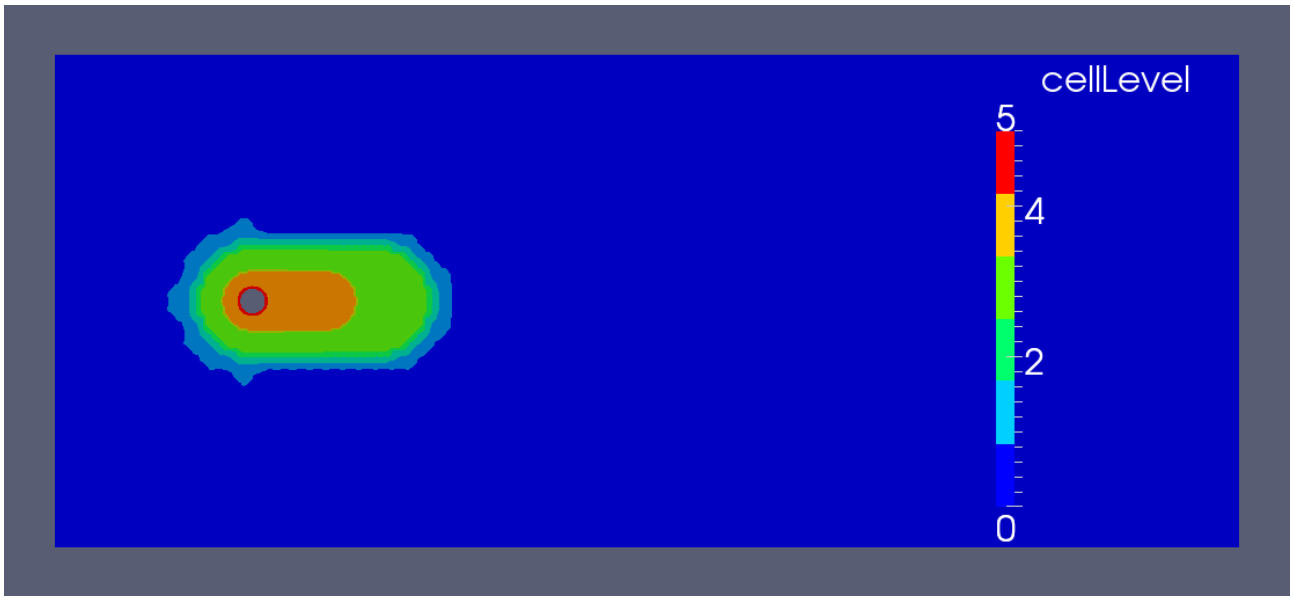

Illustration 19: Distribution of refinement levels

The cells outside the boundary layer grid were hexahedral and cubic, and refined to different levels depending on their closeness to the body. The column Δ_{focus} in Table 2 gives the cell spacing in the focus region, or near-wake. This region, extending 2.5 chord lengths or diameters downstream of the body, was refined to level 4, and the far wake, extending another two reference lengths downstream, was refined to level 3. Outside of these refinement regions, the grid was allowed to relax back to its unrefined cell spacing.

3.4 – Numerical schemes and solver settings

The simulations were run as IDDES simulations with the Spalart-Allmaras turbulence model. The pressure and velocity fields were coupled by the PIMPLE algorithm, implemented in OpenFOAM's pimpleFoam and pimpleDyMFoam solvers. PIMPLE is a hybrid PISO/SIMPLE algorithm which adds an outer correction loop to improve stability at high CFL-numbers. Since these were kept well below one in all simulation runs, this iteration loop wasn't used, and the solvers were run in pure PISO mode. For the grids on the NACA-foils, there were several skewed and non-orthogonal cells at the trailing edge. Two non-orthogonality corrector loops were therefore added to the procedure in order to partly compensate for this. The pimpleFoam solver was used for the cylinder and validation cases, and the pimpleDyMFoam solver was used for the cases involving mesh motion.

In the LES regions of a hybrid simulation, upwind differencing of the convection schemes is not a requisite for stability as it is in the RANS regions, and in order to avoid numerical diffusion it's recommended to use centered schemes on all terms in the LES region if possible [18]. Using different schemes in different regions isn't supported yet in OpenFOAM, so the convection terms were approximated with second-order upwind differencing over the whole flow domain. Time was advanced with a second-order backwards Euler scheme, and the diffusion terms were approximated with a second-order centered scheme.

3.5 – Flow conditions

The flow fields were initialized with a constant, non-zero eddy viscosity field which was replenished through the inflow boundary, meaning that the cases were run as fully turbulent. The kinematic viscosity of the fluid was set to give a Reynolds number, Re_c = 1e5 over the body.

The farfield boundaries used the *freestream* condition, a mixed velocity/pressure boundary condition that maintains a constant mass flow rate corresponding to the given velocity across all freestream boundaries. The *nutLowReWallfunction* is a simple placeholder, equivalent to a *fixedValue* condition, but which allows the calculation of *yPlus* at the wall. The *empty* conditions on the front and back planes of the RANS simulations signal to the solver that the simulation is two-dimensional, and that these faces aren't part of the solution.The *cyclic* condition on the IDDES simulations couples the front and back planes so that their values are equal everywhere. This constrains the maximum wavelength of any spanwise phenomena, but not their phase, and is therefore less limiting than a symmetry condition.

IDDES simulations			
	Farfield	Wall	Front and back planes
Velocity	*freestreamVelocity, value: (1 0 0)*	*fixedValue/ movingWallVelocity, value: (0 0 0)*	*Cyclic*
Pressure	*freestreamPressure*	*zeroGradient*	*Cyclic*
nuTilda	*fixedValue, value: 5E-5*	*fixedValue, value: 0*	*Cyclic*
nuSgs	*calculated*	*nutLowReWallFunction, value: 0*	*Cyclic*
RANS simulations			
Velocity	*freestreamVelocity, value: (1 0 0)*	*fixedValue/ movingWallVelocity, value: (0 0 0)*	*Empty*
Pressure	*freestreamPressure*	*zeroGradient*	*Empty*
nuTilda	*fixedValue, value: 5E-5*	*fixedValue, value: 0*	*Empty*
nut	*calculated*	*nutLowReWallFunction, value: 0*	*Empty*

Table 3: Overview of boundary conditions

4 – Results and discussion

4.1 – Validation & verification

A number of simulations were run in order to verify that the computational setup, with boundary conditions, solver settings and numerical schemes, was appropriate, and that the grid was fine enough to resolve the energy-carrying motions of the flow at the given Reynolds number. As mentioned in the previous chapter, two well-know flow cases were chosen for this validation: the flow over a NACA-0012 foil in and out of stall, and the crosswise flow over a circular cylinder.

The first case was chosen for its relevance to the subject under study, the flow over a rotating fin. Good results here would give confidence in the simulations' ability to function in RANS mode, i.e. to predict the forces and moments on a body in attached or barely detached flow. The proposed redesign of the turbine assembly's top fin is also a symmetric four-digit NACA-foil, though somewhat thicker, and attached flow is expected at certain angles of attack.

The second case was chosen in order to test the simulation's ability to predict the flow over a blunt object without sharp edges that would make for obvious separation points. The current top fin design has such a shape, so properly simulating this type of flow would be important for predicting the forces on the fin.

The validation cases were run for 200-400 time units, depending on the time required for the mean flow values to converge. The first twenty or so time units of the simulations were discarded. Because the LES regions of the flow were implicitly filtered, and the destruction term of the SGS model depended strongly on the cell spacing, grid convergence wasn't expected, but rather solutions that weren't overly sensitive to changes in grid resolution. Refinement would sometimes move the results away from their "target" values, showing that the relationship between the fineness of the grid and the "goodness" of the solution wasn't always straightforward for this type of simulation.

4.1.1 - The NACA-0012 foil

	90			45			10	
	C_d	C_l	Corr.	C_d	C_l	Corr.	C_d	C_l
2D URANS	2.96	0.10	2.58/0.10	1.68	1.68	1.54/1.54	0.04	0.95
Coarse	2.39	0.10	2.13/0.10	1.16	1.15	1.09/1.08	0.074	0.83
Medium	2.34	0.10	2.09/0.10	1.13	1.11	1.06/1.05	0.135	0.45
Fine	2.32	0.09	2.08/0.09	1.19	1.18	1.12/1.11	-	-
DNS, Re = 1000 [11]	**2.26**	**0.00**	**1.98/0.00**	-	-	-	-	-
DES, Re = 1e5 [20]	**2.13**	**0.16**	-	**1.13**	**1.14**	-	**0.03**	**0.95**
Exp., Re = 3e6 [1]	-	-	-	-	-	-	**0.01**	**1.10**

Table 4: Validation of drag and lift for NACA-0012

Table 4 summarizes the results from the validation runs done on the NACA-0012 foil. Mean surface pressure distributions and an α-sweep running the whole range of angles of attack are presented in the figures below. The URANS simulation was run on a two-dimensional grid made from the front face of the *"Medium"* grid. As the two-dimensional simulation was intended as a baseline case, a grid convergence study wasn't performed here, and comparisons with previously published results were instead relied on to establish its validity.

Two computer simulations and one experimental reference were used as validation cases. The DNS by Najjar & Vanka [11] was of a flat plate at Re = 1000, but as the plate was normal to the direction of the flow, the Reynolds number and geometry dependence was low, so the situations are comparable. For the DES runs at α = 10°, the flow separated, which is why the lift coefficients recorded here were so low. These flow fields tended to converge to one of two slightly different solutions, with different values for the lift and drag, though both were wrong. An investigation into why this happened is presented in the next section. Given the poor results of the DES runs at this angle of attack on the two coarser grids, it wasn't considered worthwhile to run the case on the fine one.

Like the present simulations, the DNS by Najjar & Vanka was run on a computational domain with a finite crosswise extent. They therefore used Maskell's correction formula (1) to correct for blockage effects on the drag measurements. Their domain had a blockage ratio *S/A = 1/16*, while in the present simulations, the blockage ratio *S/A = 1/20*.

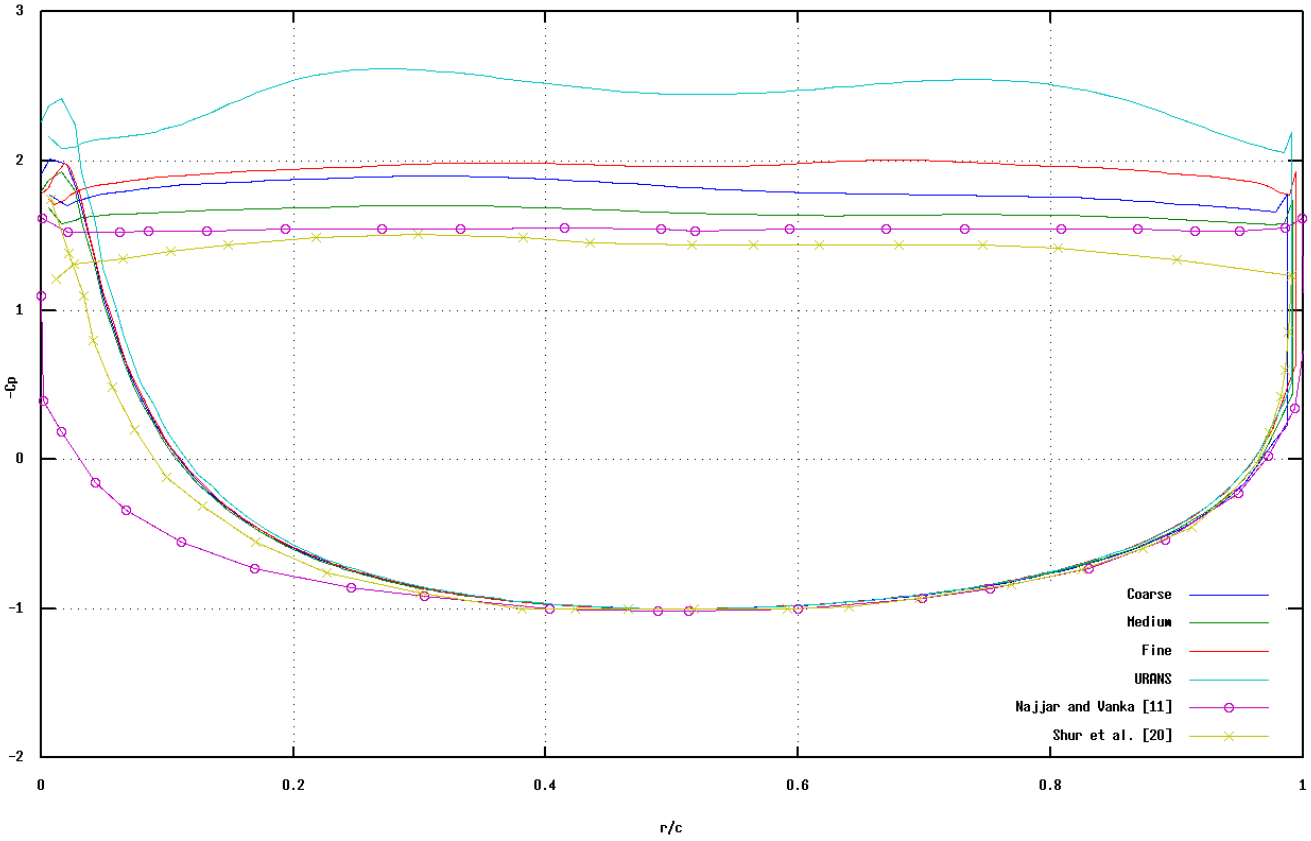

Illustration 20: Time-averaged pressure distribution over the NACA-0012 foil at 90 degrees angle of attack.

The corrections were applied to the drag and lift measurements for the foil at α = 45° and 90°, and these corrected values can be found in the rightmost of the three columns for each angle of attack. In their study, Shur et al. [20] didn't mention their results being corrected for blockage, but as their simulations were performed on an O-grid with a diameter of 30 chord lengths, a correction was probably not necessary.

Illustration 20 shows the time-averaged pressure distribution on the foil at α = 90°. As the figure shows, the URANS captured the flow on the pressure side well, but not that on the massively separated suction side, where the tight correlation between the turbulent stresses and the mean flow strain tensor disappeared. Because the URANS simulation was two-dimensional, it only captured the primary vortex shedding mode, which was in the spanwise direction. The two bumps in the time-averaged pressure distribution of the URANS case were due to these vortices being shed alternately from the leading and trailing edges of the foil.

The DES simulations also captured the other modes, which included streamwise and crosswise vorticity across a broad frequency spectrum. The mutual interference between these modes tended to flatten the time-averaged pressure distribution on the suction side and give a general reduction in pressure drag, compared to the two-dimensional case.

Note that, unlike in Najjar & Vanka's flat-plate DNS , the time-averaged pressure distribution on the suction side of the foil wasn't completely flat for the current DES simulations. Increasing the averaging period might have flattened it even further, but it also reflects a difference due to geometry, which is also present in the results of Shur et al [20]. A difference can be seen on the pressure side as well,

where the pressure distribution was slightly asymmetrical because of the thicker, rounded leading edge of the foil. It's because of this asymmetry that the foil had a non-zero mean lift coefficient even when in the vertical position.

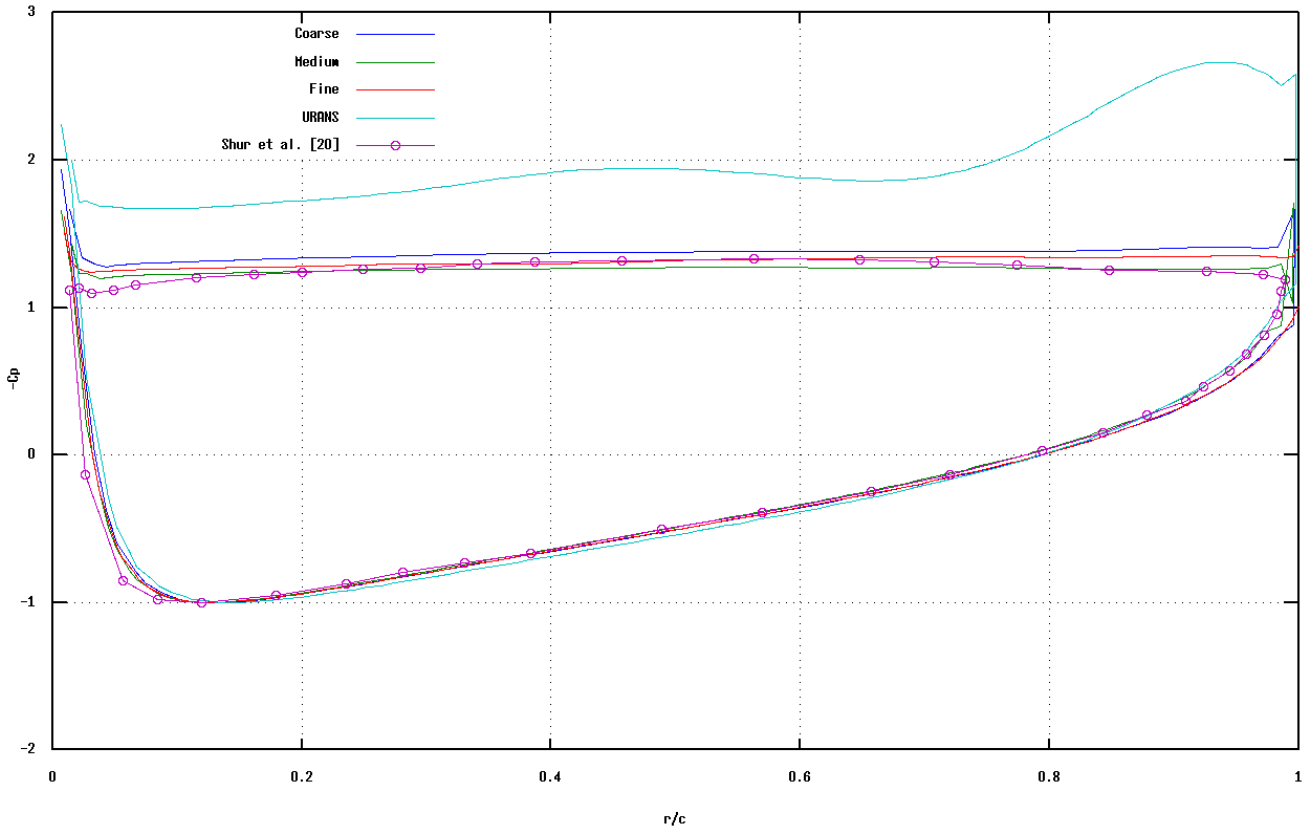

Illustration 21: Time-averaged pressure distribution over the NACA-0012 foil at 45 degrees angle of attack.

At α = 45°, the stagnation point had shifted towards the leading edge, and the pressure difference between the pressure and the suction side was less than in the vertical case, leading to a reduced pressure drag on the foil. The flow was still massively detached, so the mean pressure profile on the suction side was quite flat. Once again, the URANS simulation failed to capture this aspect of the flow, which lead to it overestimating the drag on the foil. At this angle of attack, the vortex shedding from the trailing edge was a lot more pronounced than that from the leading edge, which is reflected in the greater pressure deficiency of the URANS simulation there. This profile was validated with the DES results of Shur et al. [20], which were also from a NACA-0012 foil. The comparison shows the DES simulations matching closely on all three grids, with the *"Medium"* grid coming closest to the validation data.

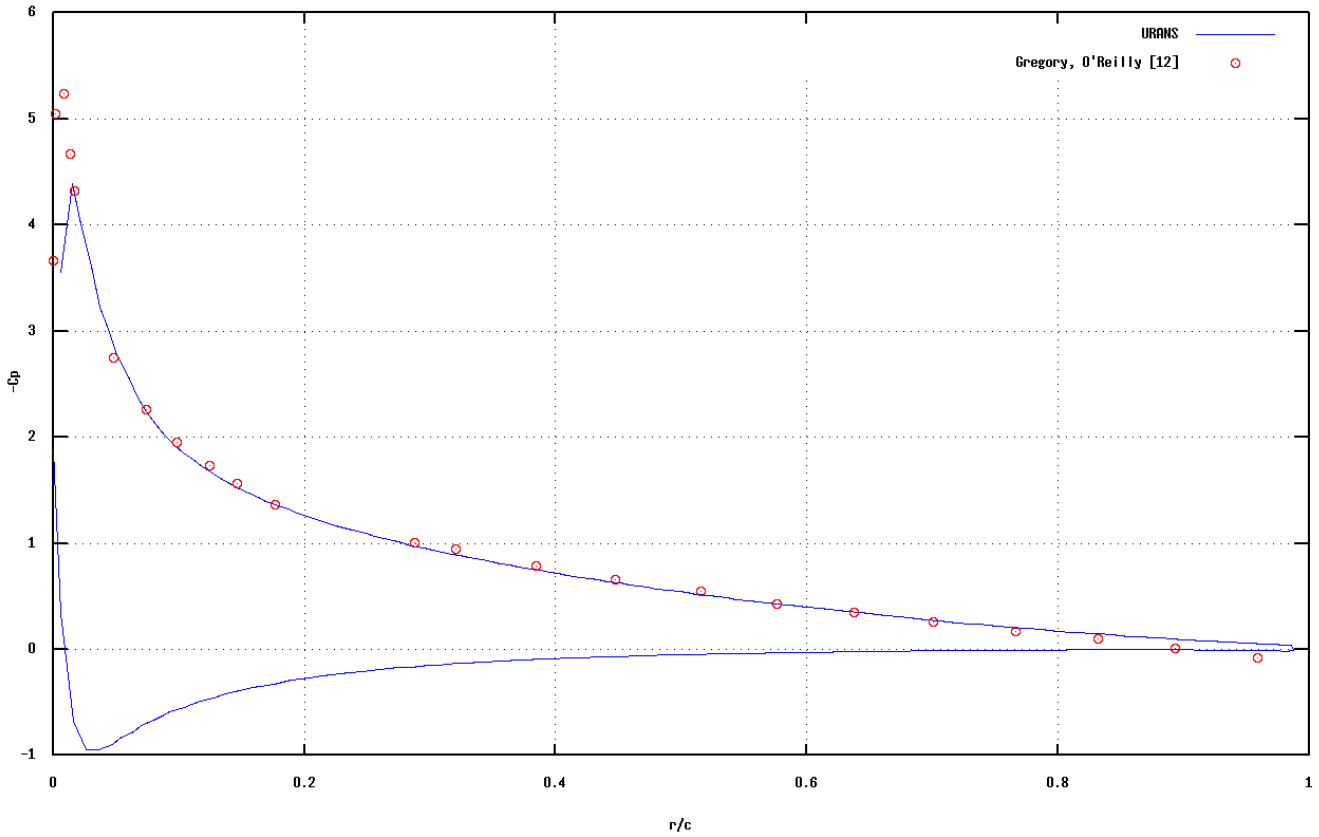

Illustration 22: Pressure distribution over the NACA-0012 foil at 10 degrees angle of attack.

At ten degrees, the URANS simulation was the only one to give reliable data. The flow here was attached, and therefore essentially two-dimensional and steady, as can be seen by the close match between the simulation data (instantaneous in this case) and the validation data on the suction side of the wing. The validation data came from experiments performed at NASA in 1970 by N. Gregory & C.L. O'Reilly, and is available at the NASA turbulence modeling resource [12].

The reason for the underestimation of the lift coefficient is probably that the sharp pressure drop right at the leading edge wasn't captured. A very high streamwise resolution would be required to trace the pressure curve here, and the grid points should therefore ideally be clustered tightly around the leading edge. A tight clustering of points on the trailing edge as well would probably have given less "smearing" of the wake, and a drag coefficient more in line with experimental data. Some of the difference can also be attributed to the difference in Reynolds number between the simulation and the experiment.

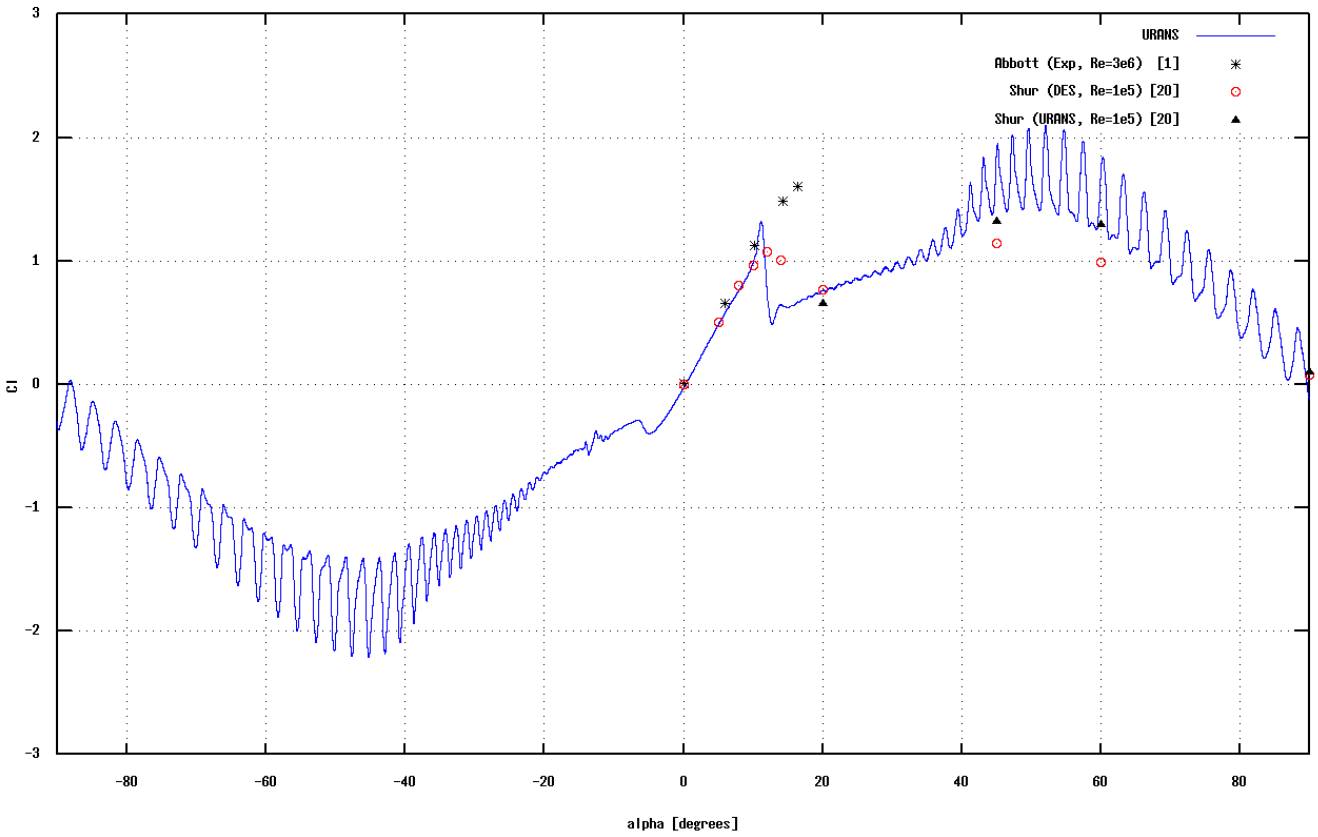

Illustration 23: Time trace of the lift coefficient on a NACA-0012 foil during an α-sweep.

The NACA-0012 foil was taken through a frequency sweep, a 180° rotation from a -90° to a 90° angle of attack. The rotational speed was one half of a degree per time unit. This gave the trailing edge of the foil a velocity $U = 7.85e\text{-}3 * U_{inlet}$. No sensitivity study was done on the rotational speed, but an analysis of the frequency spectra indicated that it was of minor importance in this simulation. The argument can also be made that since the vortex shedding frequency was in the range $St = 0.2\text{-}0.4$, the foil rotated at most 2.5 degrees on a shedding cycle, indicating that the rotation had a small effect on the flow field.

Illustration 23 above gives the variation of the lift coefficient as the foil was taken through its rotation. As indicated by the validation results above, two-dimensional URANS simulations have a tendency to overestimate the forces on a body in massively separated flow. An approximate measure of by how much the mean forces were overestimated can be had by comparing with the data of Abbott and von Doenhoff and Shur et al. which is plotted together with the curve [1, 20].

The foil started out in the vertical position with a small mean lift component, due to the foil shape being not quite symmetrical about the horizontal plane. As the foil turned down towards $\alpha = -45°$, the vertical component of the pull from both the leading edge and trailing edge vortex became of the same sign, and a mode at approximately double the frequency started making itself felt. At $|\alpha| = 45°$, the lift coefficient had found both its maximum mean value and its maximum oscillation amplitude. After this point, both the mean value and the oscillations decreased rapidly. Their frequency increased as the oscillations were damped out, until they eventually disappeared near $\alpha = -10°$.

Note the asymmetry of the lift curve about the middle, due to a hysteresis of the flow. The flow attached at $\alpha = -5°$, and detached again at $\alpha = 11°$. Abbott & von Doenhoff [1] registered attached flow on the NACA-0012 foil until about $\alpha \approx 16°$, but their data were from experiments performed at much higher Reynolds numbers. The DES data of Shur et al. [20] show separation at about the same angle as seen here. The symmetry seemed to be regained for $|\alpha| > 45°$.

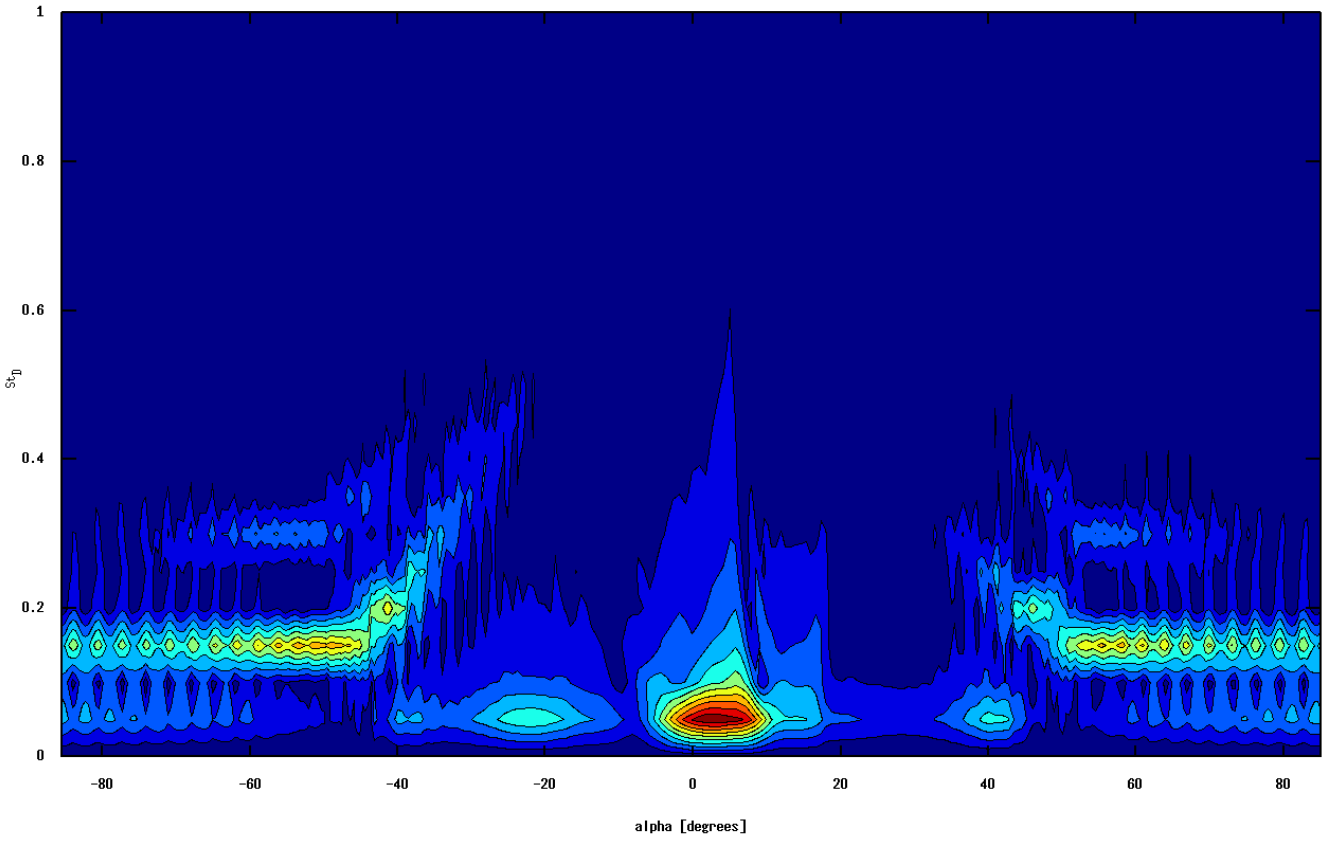

Illustration 24: Frequency spectra of the lift coefficient on a NACA-0012 foil during an α-sweep.

The frequency spectra of the force coefficients at different angles of attack were found by cutting the data into samples of equal length and applying a discrete Fourier transform on each sample. A compromise had to be made here between the length of each individual sample and the frequency resolution of their Fourier transform. The discrete Fourier transform of each sample contained a number of data points equal to the sampling frequency times the duration of the sample. The range of frequencies covered by the transform was [*-sampleFreq/2:sampleFreq/2*]. This gave a frequency resolution which was equal to the inverse of the sample duration. Choosing a long sample duration would therefore give a high frequency resolution, but at the cost of "smearing" the data out along the α-axis.

A sample duration of twenty time units was therefore chosen, in order to be able to resolve the frequency to within $St_D = St \pm 0.025$, with St_D the discrete and St the actual dimensionless frequency. The foil then turned ten degrees in the course of one sample, which was found to be acceptable. The samples were staggered one time unit apart, giving 340 individual samples over the course of a frequency sweep. The "DC" component of the signal was removed and the amplitudes were normalized with respect to the highest amplitude in the data set. The spectra were then plotted as filled contours at ten equally spaced levels, as seen in Illustration 24, representing the spectra of the lift coefficient on the NACA-0012 foil.

For $|\alpha| > 45°$, the principal mode of oscillation was found at $St_D = 0.015$. At $\alpha \approx -70°$, the second harmonic of the principal mode made its appearance, due to same-sign force contributions from vortices at both ends of the foil. This can be seen as the dimples in the time signal. The dampening and quickening of the oscillations for $|\alpha| < 45°$ is also represented in the spectra. The lowest row, at $St_D = 0.05$, represents the rate of growth and decay of the lift coefficient as the foil rotated, with the bright red region representing the almost linear growth of the lift coefficient in the attached flow regime. Hysteresis is also apparent, as the red dot isn't centered on α = 0°, and the spectra are somewhat quieter for α = [11°,45°].

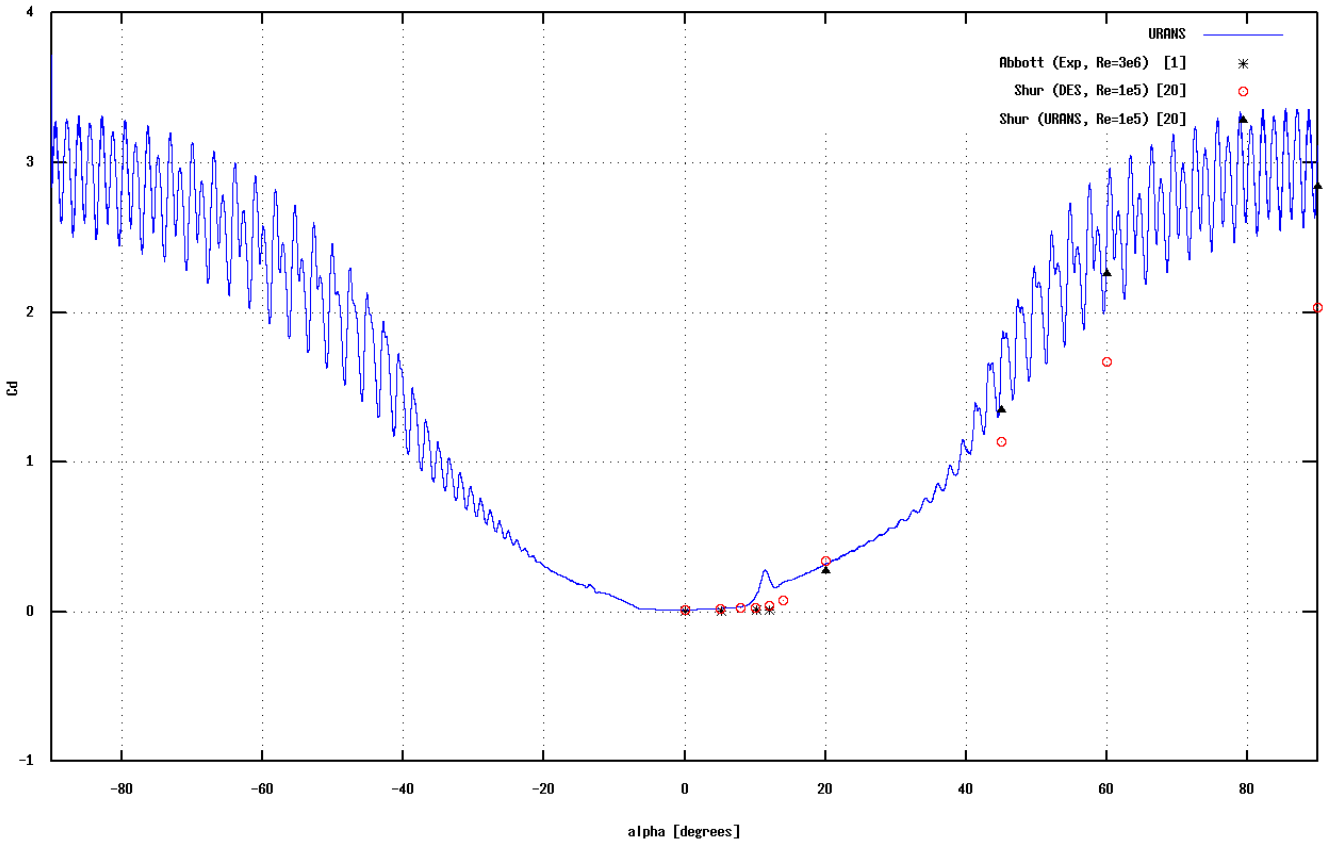

Illustration 25: Time trace of the drag coefficient on a NACA-0012 foil during an α-sweep.

Illustration 25 gives the evolution of the drag coefficient during the frequency sweep. It shows a slight overprediction of the drag compared to the validation data at low angles of attack, and a small spike in the curve where the flow separated from the foil. Both the drag and lift at $\alpha = 20°$ was well captured, and the forces on the foil were probably well predicted by the URANS simulation for angles of attack

up to about thirty degrees, where the oscillations started to become significant. As expected, URANS overpredicted the forces for massively separated flows, but the current results agreed well with the URANS results of Shur et al. [20].

As each vortex shed represented one cycle of the drag coefficient, its oscillations had double the frequency of the lift coefficient at $|\alpha| = 90°$. As the foil moved away from the vertical position, the contribution from the leading edge vortex shedding became progressively less significant, and it was almost gone for $|\alpha| < 45°$.

The spectra in Illustration 26 show that the drag oscillations at high angles of attack had, at $St_D = 0.3$, twice the frequency of the lift oscillations, but that they transitioned into the lower frequency mode at lower angles of attack.

The amplitude of the drag oscillations dropped sharply as α went from -45° to -20°, and their subsequent increase on the other side was even sharper, due to hysteresis making the α-range between flow detachment and maximum oscillation narrower. This is reflected in the two bright spots on the first row, at $St_D = 0.05$. The small spike in the drag coefficient as the flow separated at $\alpha = 11°$ is also visible.

The good agreement between the contour plot of the spectra and the time trace of the force coefficients indicates that the basic methodology and choice of sample duration is sound, and that the plot is fit for qualitative observations. The plots also show a clear separation in time scales between vortex shedding and rotation.

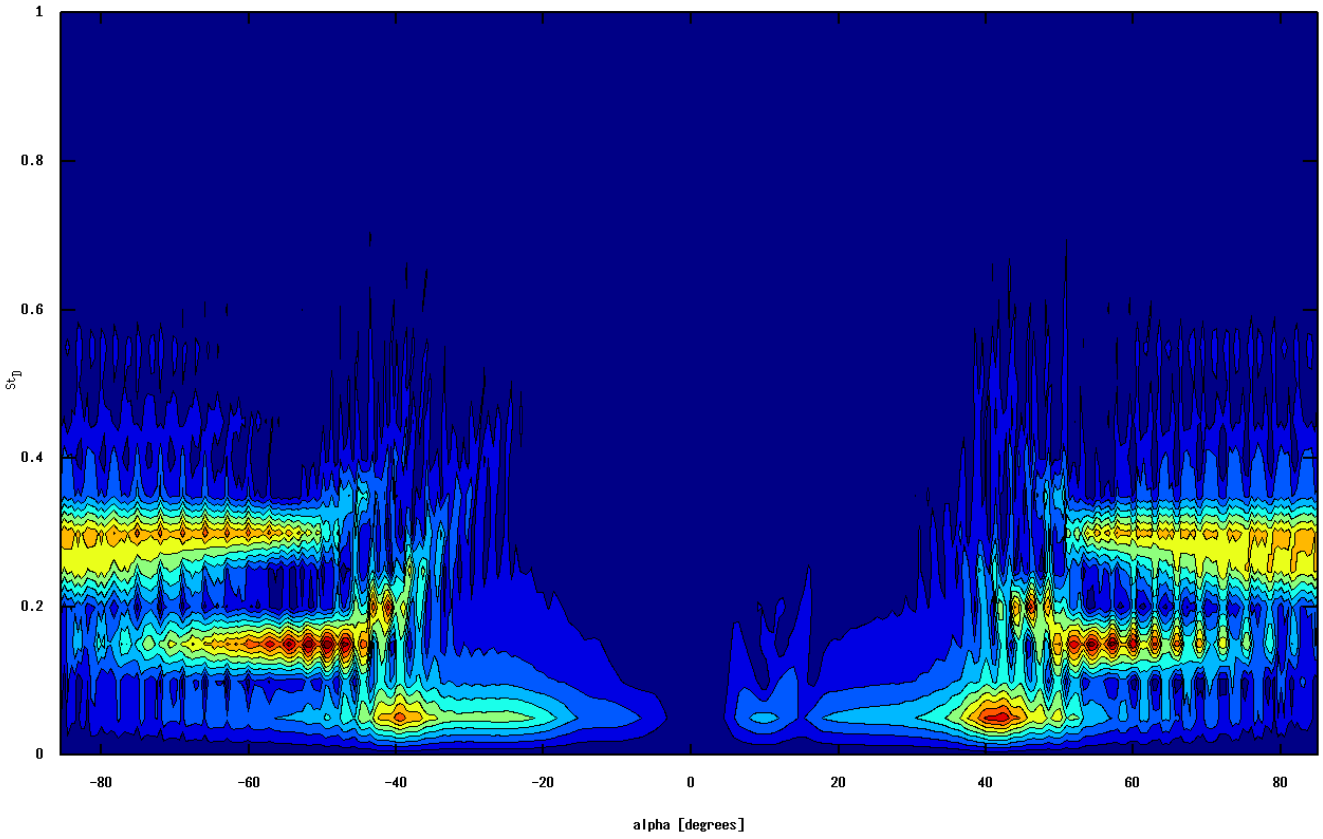

Illustration 26: Frequency spectra of the drag coefficient on a NACA-0012 foil during an α-sweep.

4.1.2 – The circular cylinder

	C_d	$C_{l,rms}$	$-C_{pb}$	St	θ_{sep}	Lr
2D URANS	1.14	0.74	1.37	0.23	85.4	0.52
Medium	0.94	0.10	0.88	0.22	79.8	1.75
Fine	1.02	0.21	1.00	0.22	79.8	1.36
DES, Re = 1.4e5, LS [18]	**0.87**	**0.10**	**0.81**	**0.23**	**78**	**1.50**
DES, Re = 1.4e5, TS [18]	**0.64**	**0.10**	**0.70**	**0.28**	**94**	**1.20**
Exp., Re = 1e5, LS [3]	**1.24**	**0.24**	**1.20**	**0.20**	**78**	-
Exp., Re = 5e6, TS [3, 16]	**0.75**	-	**0.85**	**0.28**	**110**	-

Table 5: Overview of validation results on the cylinder case

Countless studies of this flow case exist, both experimental and numerical, but the results are widely scattered, indicating that it's difficult to find reliable data for several of the flow properties. Four cases were chosen for validation, with two of them being numerical and two being experimental studies. For each type of study, there was one case with turbulent boundary layer separation, and one with laminar separation.
C_d denotes the average drag coefficient, $C_{l,rms}$ the RMS value of the lift, C_{pb} is the back pressure coefficient, St is the dimensionless frequency, θ_{sep} is the separation angle, defined as the point at which the wall-normal gradient of the mean velocity is zero. Finally, L_r is the length of the recirculation bubble, defined as the distance from the base of the cylinder to the point at which the mean streamwise velocity becomes positive.
Three cases were run for comparisons with the validation data. The first was a two-dimensional URANS, run on a grid made from the front face of the *"Medium"* grid. The other two were DES simulations, run on three-dimensional grids of different fineness. As the cases were all run with non-zero eddy viscosity at the inlet, the boundary layers were expected to be fully turbulent. That the results seemed to agree so well with the laminar separation cases instead was therefore a surprise.
As the table shows, there was some disagreement between the two DES runs, especially in what concerned the forces on the cylinder and the length of the recirculation bubble. A wider grid refinement study might therefore have been appropriate, but the time trace of the force coefficients and the wide scatter of the validation data suggested that there were other sources of error that were as significant.

The time-averaged pressure and skin friction distributions are given in Illustration 27 below. Here as well, the data clearly matches the laminar rather than the turbulent separation cases, especially in the maximal amplitude and crossover point of the skin friction curve.

The pressure and the wall-normal velocity gradient on the upstream side of the cylinder surface showed strong streamwise fluctuations that didn't move or weaken noticeably with further averaging, so it's possible that they represented a standing pressure wave on the surface, maybe due to a boundary layer instability leading up to the separation. These fluctuations are reflected in the zigzag curve of the pressure and skin friction distributions. Apart from the fluctuations, the pressure and skin friction data matched Achenbach's [3] experiments quite well, although the pressure dropped off a bit near the back of the cylinder in the URANS simulation. This is reflected in the higher measured mean drag of that case.

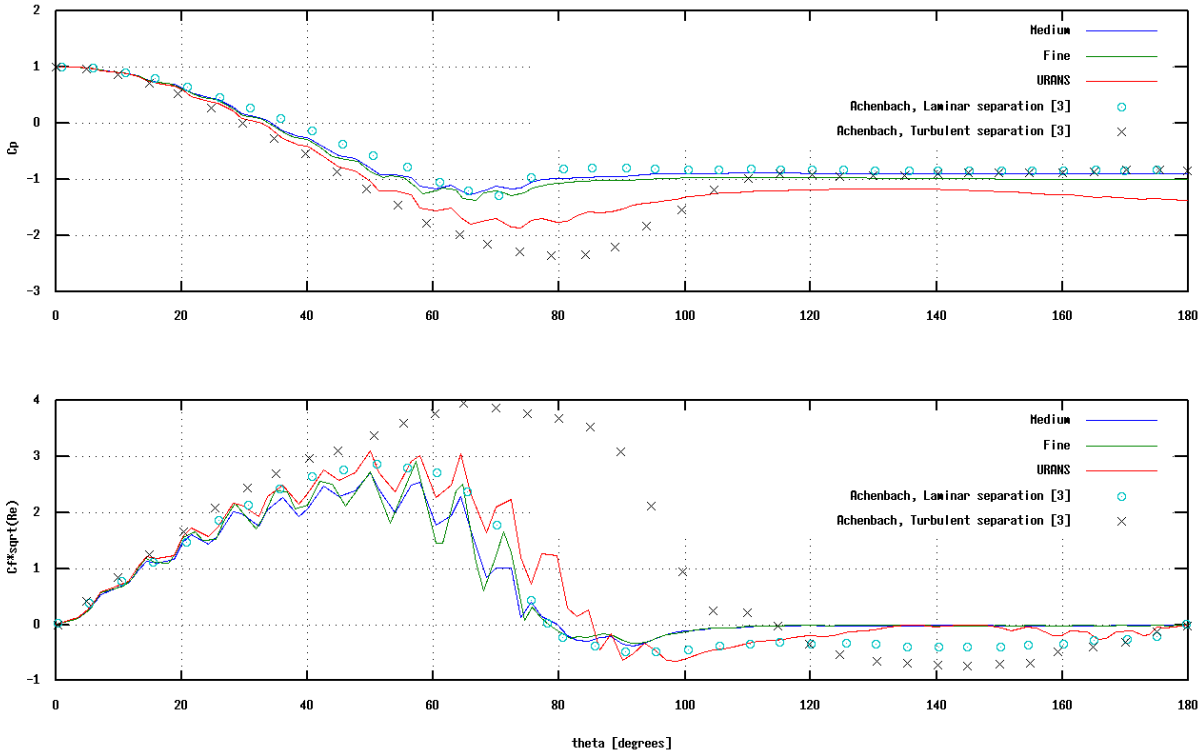

Illustration 27: Time-averaged pressure (top) and skin friction (bottom) distributions over the circular cylinder

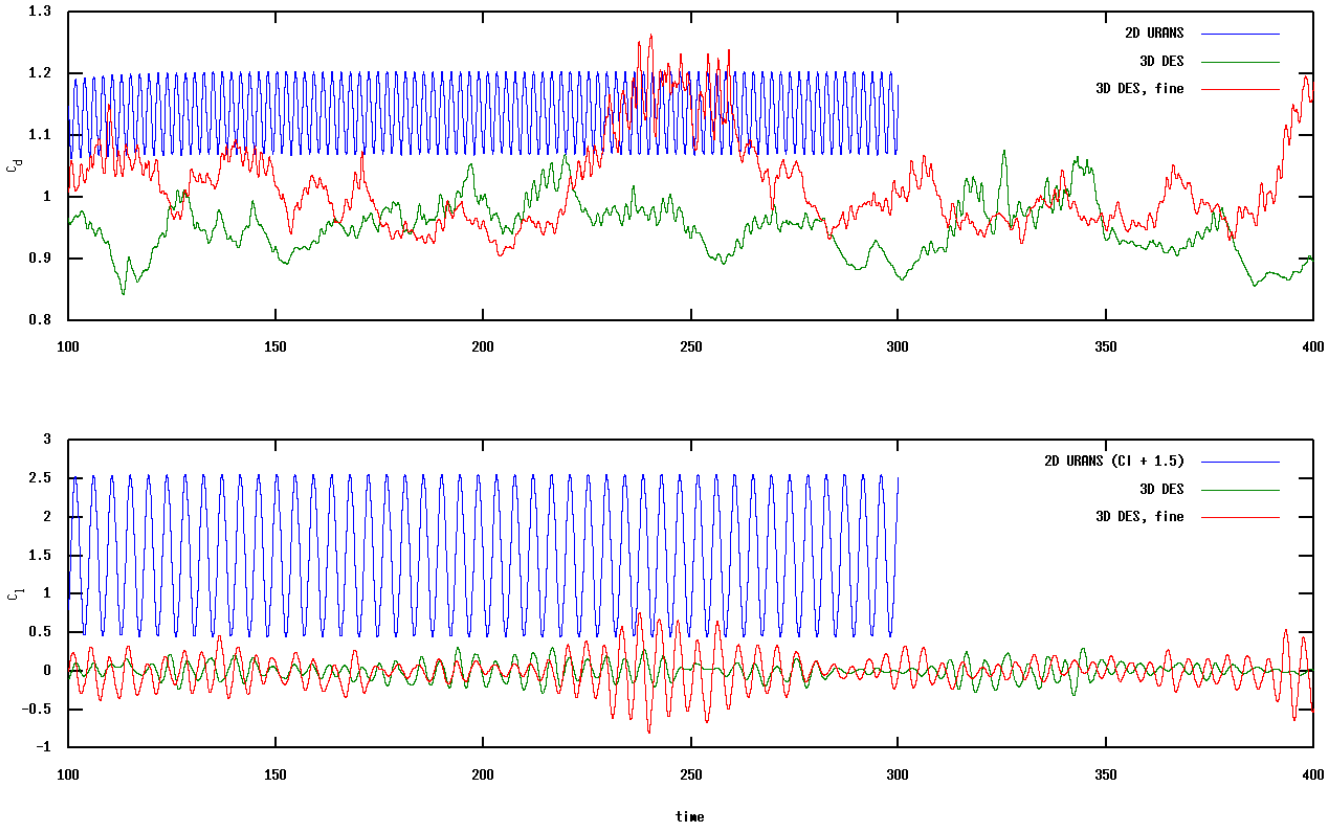

Illustration 28: Time traces of drag and lift on the three circular cylinder cases.

The time traces of the drag and lift coefficients for the URANS case and the two DES cases are shown in Illustration 28. The lift coefficient curve of the URANS case has been shifted by 1.5 units for better visibility. For the three-dimensional cases, the drag coefficient seemed to be almost completely uncorrelated to the vortex shedding frequency, and was instead dominated by low frequency noise. It was hard to see a periodicity of the signal, even over 400 time units.

As was also noted by Shur et al. [18], the force oscillations seemed to go through periods of high and low activity, and the drag and lift variations were tightly coupled in this regard. The DES simulation on the fine grid might have been in some long-lasting supercycle of generally high activity, with many excursions, although it seemed on average to show higher forces than the one run on the medium grid.

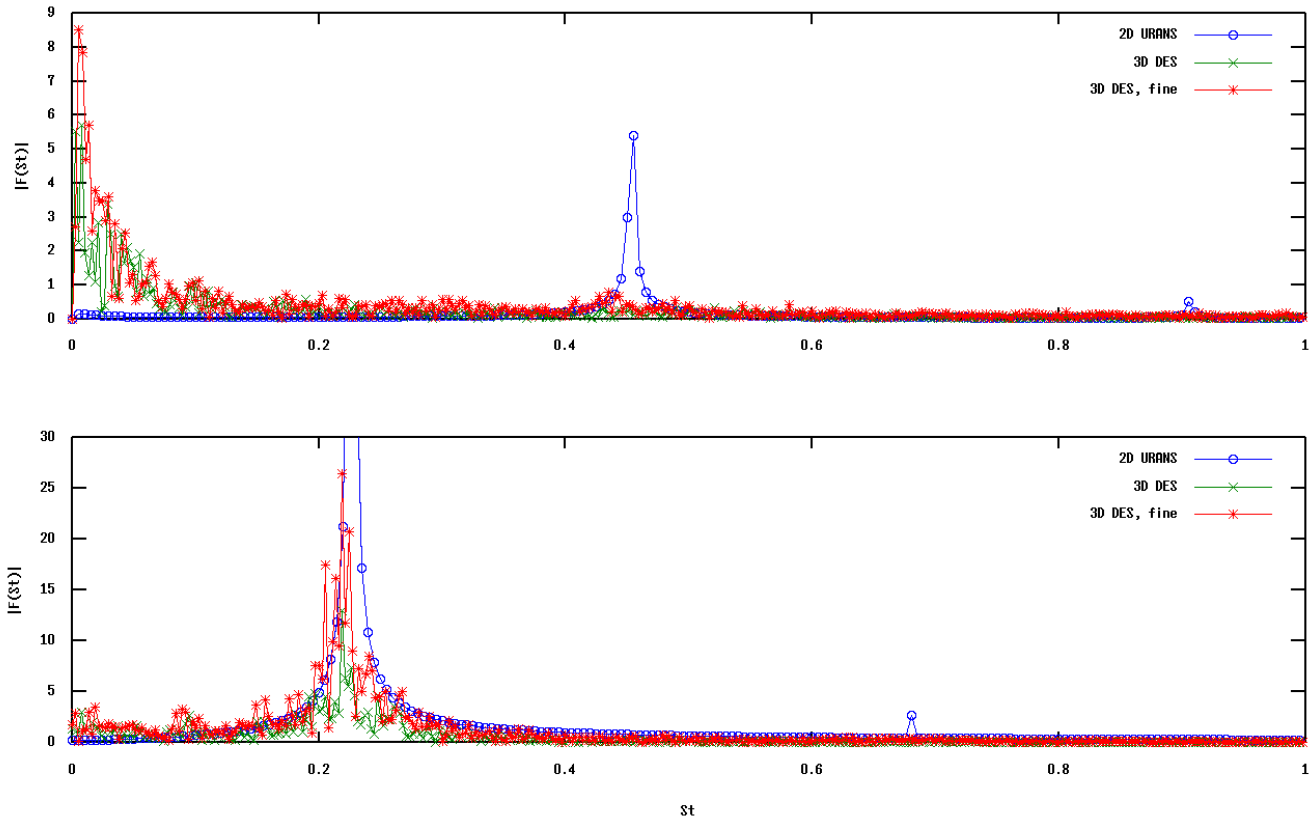

Illustration 29: Frequency spectra of drag and lift on the three circular cylinder cases

The lift coefficient gave a much cleaner signal, and the shedding of the spanwise vortices can easily be distinguished. The signal was modulated by a few lower frequency modes, and the switching between generally high and low activity can also be seen. The URANS captured the frequency of the primary vortex shedding very well.

The frequency spectra of the lift and drag coefficients can be seen in Illustration 29, and they clearly show the low frequency of the drag signal for the DES simulations. Very little energy can be found at the frequency associated with the primary vortex shedding, and the dominant modes of the drag spectrum are mostly at $St < 0.1$. The frequency resolution of the FFT of the DES simulation on the fine grid was $\delta St = 2.70e\text{-}3$, and the peak of that spectrum was found at $St_{peak} = 5.41e\text{-}3$. The most prominent mode therefore had a period of $St^{-1} = 185$ time units.

As a result, the convergence of the mean drag coefficient was slow, as can be seen in Illustration 30. Long simulation times were therefore required to establish precise values for this and related quantities, like the recirculation length and back pressure coefficient.

The lift coefficient spectra were all centered around the primary vortex shedding frequency, with the signals from the DES simulations modulated by some weaker low frequency modes. The dominant mode in the URANS simulation

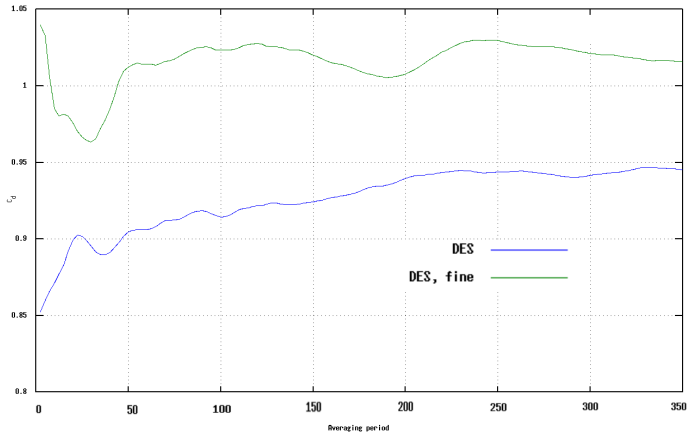

Illustration 30: Evolution of mean drag coefficient with averaging period, given in non-dimensionalized time units.

had an amplitude $|F(St_{peak})| = 88$, so the vertical axis was cut low. The cause of the second little peak in the URANS' spectra is not known.

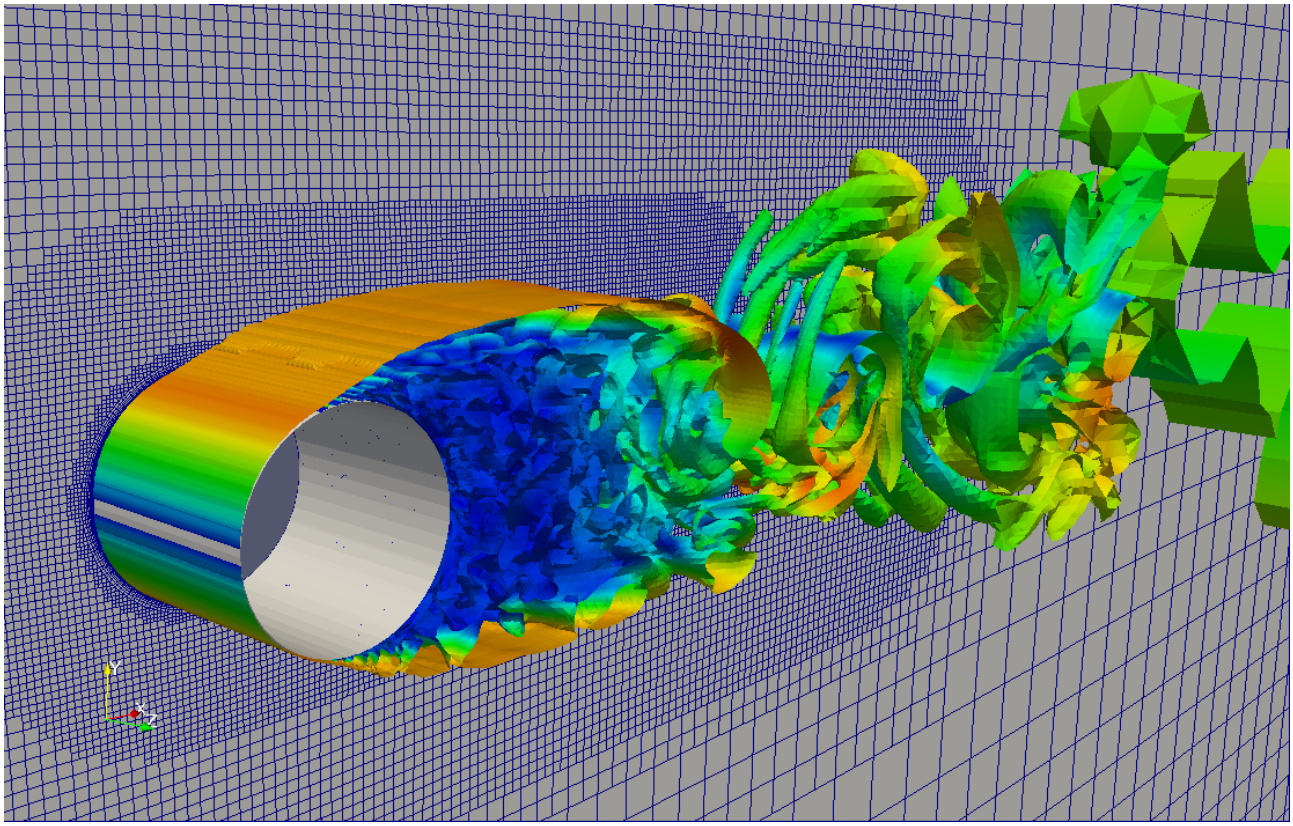

Illustration 31: Isocontour of Q = 1e-5 from a DES of the flow over a circular cylinder on the "medium" grid.

Illustration 31 shows a vortex structure in the form of a smooth, nearly two-dimensional sheet of high-velocity fluid rising up from the separation point at the cylinder surface. It continues unbroken for some distance down the wake, rolling up into the vortex which has been shed from the cylinder's top half some moments earlier. Faint ripples can be seen in the sheet in the streamwise direction, which is the beginning of a temporally evolving instability that will eventually break up the sheet. Though not clearly visible in the picture, the sheet at the bottom of the cylinder has already begun to unravel, and is retreating back to the cylinder in preparation for the next vortex.

Streamwise rollers form S-shaped curves that connect the outward-facing halves of consecutive spanwise vortices. Several of these rollers fit side by side in the domain, an indication that they may not have been overly constrained by its spanwise extent. The sheet, on the other hand, might have been less stable given a wider domain.

The recirculation region, coloured blue in this picture, is dominated by small-scale, chaotic motions. It's probably these motions that are the cause of the broad, low-frequency noise in the drag spectrum. The lift spectrum seemed much less affected by this randomness. As the force coefficients represent an averaging over the whole cylinder, it was probably the symmetry of the recirculation region about the streamwise plane that caused these random motions to largely cancel each other out in the lift signal. As the grid coarsens downstream, the structures degenerate into something resembling the simple two-

dimensional vortices found in the RANS solution.

4.1.3 – Summary of the validation

The grids, numerical schemes and time step seemed appropriate for simulating massively separated flow at this Reynolds number. Grid refinement didn't reveal a strong grid dependence of the results, and the *"Medium"* grid resolutions will be used for the remaining simulations. The RANS part of the DES simulations seems to have failed, and an investigation into that issue will be presented in the following section.

The flow over a circular cylinder was reproduced quite well, although the flow characteristics indicated that the boundary layers were laminar rather than turbulent on separation. At the low Reynolds numbers simulated here, laminar separation would be more realistic in any case.

The URANS simulations, which were much less demanding in terms of computing time than the DES simulations, predicted the forces on the NACA-0012 foil quite well, even for angles of attack up to about 30 degrees from the horizontal. They also successfully captured the primary vortex shedding mode in massively separated flow, but tended to overestimate the forces on the objects.

Particularly the force oscillations seemed to be strongly over-predicted by the URANS. The DES simulations unlocked lower frequency modes which dampened them quite drastically. The two-dimensional alpha-sweep showed a neat gap in time scales between the vortex shedding from the foil and the modulation of the signal due to the rotation of the foil, but this gap might be made narrower by the unlocking of these new vortex shedding modes, making it more difficult to separate the contributions from each.

4.2 – An investigation into the modeled stress depletion of the DES simulations

As the validation simulations showed, the flow fields around the foil in deep stall were reasonably accurate. The results at $\alpha = 10°$ were far off the mark, however, showing a flow that couldn't stay attached to the foil even at an angle of attack that should have been well within its operating range. The two-dimensional RANS-simulation that was performed using the front face of the *"Medium"* grid had no problems with premature separation, and gave reasonably accurate values for both the lift and drag forces, with no signs of unsteadiness or vortex shedding.

Something similar can be seen for the simulations of the flow over a circular cylinder. The flow separated at an angle $\theta_{sep} < 90°$ even though the simulation was run with non-zero eddy viscosity at the inlet, and the boundary layer was supposed to be turbulent right from the onset. The recorded values were typical of separation from a laminar boundary layer, where the turbulence model lies dormant until the point of separation. The skin friction, surface pressure distribution, recirculation length and other flow properties all showed values indicative of laminar separation, meaning that, before separation, the turbulence model either didn't activate or only played a negligible part.

A look at the RANS/LES-indicator function for the flow field around the NACA-0012 foil confirms this impression. It shows that the simulation was only in RANS mode in a very thin layer near the wall, spanning only three or four cells in the wall-normal direction over most of the surface. There was also a lot of noise in the area immediately upstream of the body, but since the flow gradients and eddy viscosity were both low in this region, it was thought likely that it wouldn't affect the flow field.

Illustration 32: RANS/LES distribution for DES of NACA-0012 foil at ten degrees angle of attack on "medium" grid.

Plotting the indicator function against the y^+-values from the RANS simulation of the foil at $\alpha = 10°$ shows that, over much of the foil, the IDDES simulations went from RANS to LES mode in the middle of the boundary layer.

This premature switch caused modeled stress depletion, which in turn lead to grid-induced separation (GIS) of the flow. GIS is usually caused by so-called ambiguous grids, where the anisotropy of the near-wall cells is too weak to lock the DES model in RANS mode across the whole boundary layer.

Due to the "isotropic" refinement algorithm in snappyHexMesh, the boundary layer grid on the foil surface was very much overrefined in the spanwise direction, with 128 points. That on the surface of the cylinder and the top fin had 64 points. The grid spacing in the streamwise direction was the same as in the spanwise.

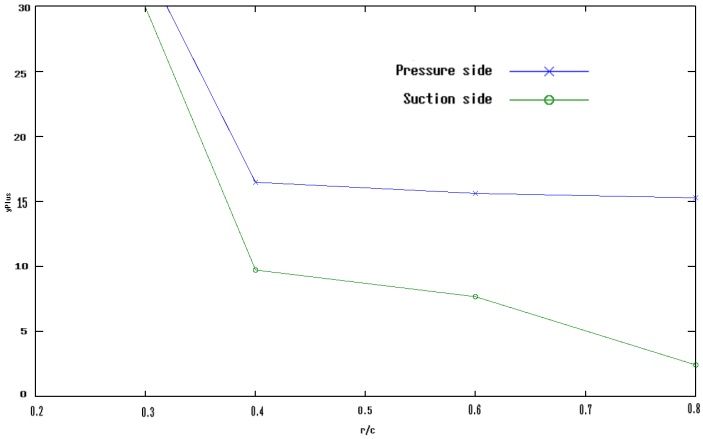

Illustration 33: Location of RANS/LES interface in wall units at several points on the NACA-0012 foil.

The boundary layer thicknesses were measured on the RANS simulation of the NACA-0012 foil at ten degrees angle of attack, taking the point at which the streamwise velocity was 95% of its freestream value as the outer edge of the boundary layer. On the suction side, near the trailing edge, the boundary layer reached a maximum thickness of $\delta_{s,max} \approx c/30$. On the pressure side, the maximum boundary layer thickness was about $\delta_{p,max} \approx c/75$.

4.2.1 - Initializing the f_d-field

This meant that the grids were indeed ambiguous, but the shielding function in DDES and IDDES should have been able to keep the model in RANS mode despite of this. However, if the grid-induced separation kept boundary layers from forming in the first place, the shielding function might not have had a chance to become active.

To make sure that the boundary layers were present from the start, and that the shielding function would have a good initial distribution, the flow field was initialized with the RANS solution. The stock implementation of IDDES was used. Unfortunately, the initial condition was quickly erased by the simulation, and within just a few time units, the flow field was indistinguishable from one initialized with a uniform velocity. This indicated that the problem wasn't about getting into the right basin of attraction, but that attached flow over the foil was a solution that the flow field couldn't converge to given the current case setup.

4.2.2 - Increasing the anisotropy of the near-wall grid cells

With a sufficiently coarse wall-parallel grid spacing, the DES should stay in RANS mode over the whole boundary layer, regardless of the value of the shielding function. In order to increase the value of Δ near the wall, and thereby the reach of the RANS region, the grid was extruded to a spanwise resolution of 32 cells. This increased the spanwise grid spacing on the foil by a factor of four.

Because of a problem with the face matching in the *createPatch* utility, it wasn't possible to regenerate the cyclic patches after extruding only a part of the grid. The entire grid therefore had to be extruded to an even spanwise resolution of 32 cells. This represented a waste of grid cells in the departure and Euler regions, but the gains in the RANS region compensated for this to the extent that the uniformly extruded grid over the NACA-0012 foil actually had fewer cells than the original *"Medium"* grid.

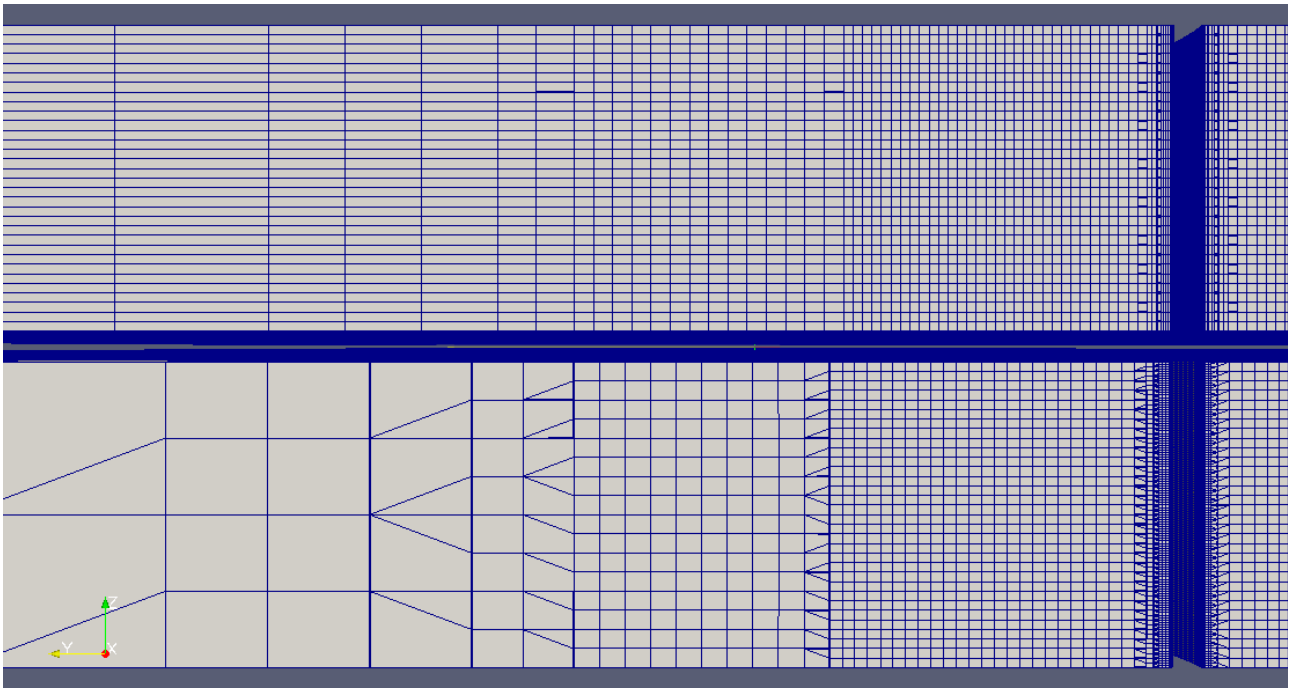

Illustration 34: Sideviews of uniformly extruded (top) and original (bottom) grids.

As Illustration 35 shows, the extrusion didn't have any visible effect on the distribution of RANS and LES mode in the flow field, indicating that the ambiguous grid wasn't what caused the Grid Induced Separation. The distribution of RANS and LES cells near the wing didn't seem to depend on the wall distance at all.

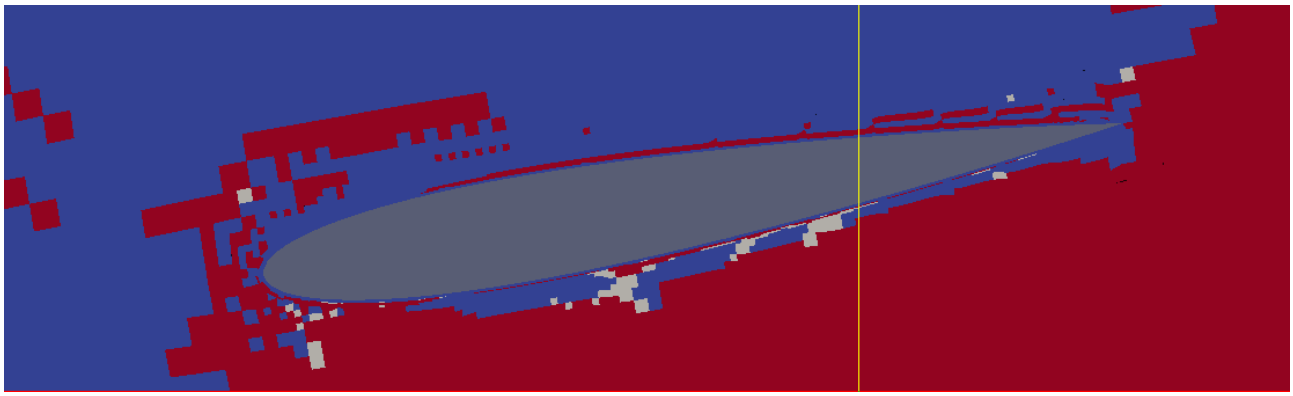

Illustration 35: RANS/LES distribution for DES of NACA-0012 foil at ten degrees angle of attack on uniformly extruded grid.

4.2.3 - Leaving the variable Δ-definition

As mentioned in section 2.4, the sub-grid length scale Δ was redefined for IDDES in order to make it more suited for wall-modeled LES. This redefinition meant that in the cells very near the wall, the sub-grid length scale Δ_{IDDES} was equal to $C_w h_{max}$, a fraction of the value from DDES and DES97. The anisotropy of the near-wall grid cells would therefore have to be increased by a factor $1/C_w$ in order for the grid to have the same preserving effect on RANS mode in IDDES as in the older versions. IDDES therefore seemed to rely almost exclusively on the shielding function to keep the near-wall solution in RANS mode, which was a potential weak spot.Furthermore, in going through the source code, a bug was found in OpenFOAM's implementation of the IDDES delta, in the lines corresponding to equation (18). A temporary, dummy value *tfaceToFacenMax* was used for the wall-normal cell spacing instead of the proper value *faceToFacenMax*. Since the wall-normal growth rate of the boundary layer grids in the present study was greater than $(1 + C_w)$, an IDDES simulation performed on them would use this faulty definition.

```
00111    delta_.internalField() =
00112        deltaCoeff_
00113       *min
00114        (
00115            max
00116            (
00117                max
00118                (
00119                    cw_*wallDist(mesh()).y(),
00120                    cw_*hmax
00121                ),
00122                tfaceToFacenMax
00123            ),
00124            hmax
00125        );
```

In light of this, it was decided to scrap the variable Δ-definition in favour of the old, constant one. To retrieve the original definition from DDES and DES97 , the parameter C_w was set equal to one, with $h_{max}=max(\Delta x, \Delta y, \Delta z)$. The IDDES simulation with the DDES delta was run on the extruded grid, and as Illustration 36 shows, the RANS region was now smoother and more coherent, enclosing the whole foil. There was still a lot of noise present, and the flow field was no different from before, indicating that any RANS functionality there was had only a negligible effect on the flow. The pure RANS simulations had shown that the grid was capable of resolving the boundary layer at this Reynolds number and angle of attack.

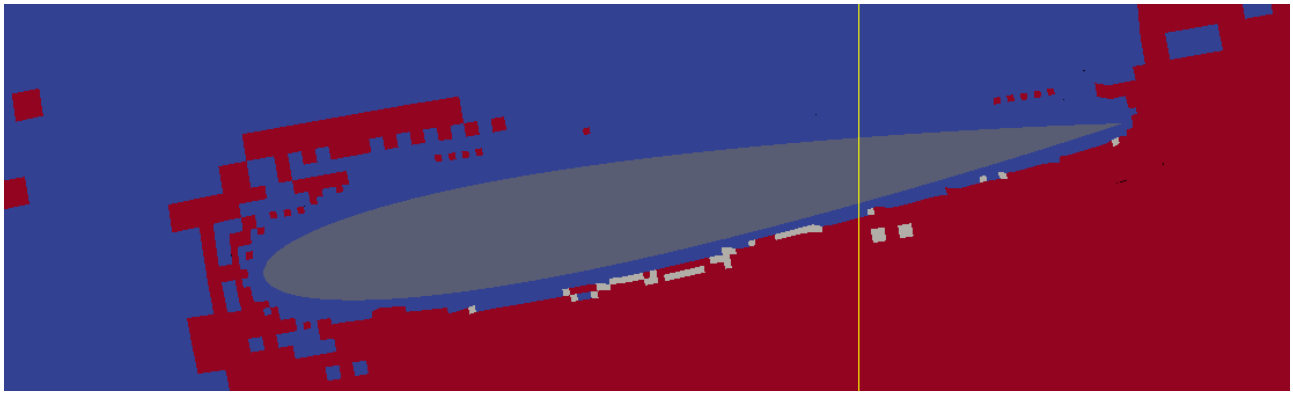

Illustration 36: RANS/LES distribution for DES of NACA-0012 foil at ten degrees angle of attack on uniformly extruded grid with constant Δ-definition

4.2.4 - Switching to DDES

Some test runs were now made where IDDES was left altogether, in favour of DDES. As Illustration 37 shows, the noise that was present in the IDDES simulations disappeared completely. DDES simulations were run on both the regular and the extruded grid, in order to see if the implementation of the shielding function in the DDES code would be more successful in preserving RANS mode than that in the IDDES code. The smooth blue ring enclosing the foil represents the cells that were kept in RANS mode by the grid, while the ragged patches of blue around it represent the cells kept in RANS mode by the shielding function. On the extruded mesh, the blue ring was four times wider, but this simply corresponded to the spanwise coarsening of the grid. The shielding function didn't seem to have any coherent area of effect in either of the two flow fields.

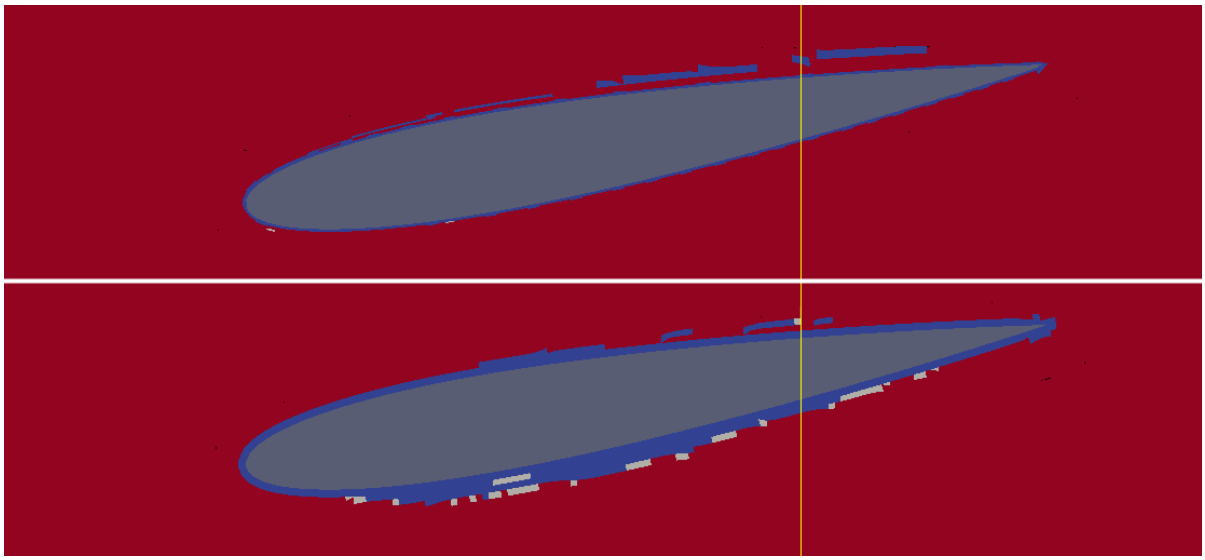

Illustration 37: Comparison of RANS/LES distributions for DES of NACA-0012 foil at ten degrees angle of attack: DDES on "medium" grid (top) and DDES on uniformly extruded grid (bottom)

For these DDES simulations, however, there might be another reason why the shielding function didn't activate properly. As mentioned in the presentation of the Spalart-Allmaras model, the $\tilde{\nu}$-destruction term depends strongly on the model length scale (wall distance in RANS mode, cell spacing in LES mode). When using this model as an SGS model in low-Re DES there's therefore the danger that a fine cell spacing in the LES region will cause the near-wall destruction term to kick in. When it has brought $\tilde{\nu}$ low enough compared to the kinematic viscosity ν, the f_{v1}-term will pull ν_t the rest of the way down to zero, effectively making the flow laminar. In IDDES, the low-Re correction term Ψ prevented this.

Illustration 38 Shows that the eddy viscosity dropped dramatically with each refinement level once the grid reached a certain degree of fineness. By the time the flow reached the leading edge, ν_t had dropped to almost zero and the flow was practically laminar. The boundary layer couldn't stay attached to the foil, and, due to a general absence of eddy viscosity, the shielding function mostly stayed dormant. This sharp turbulence decay was not seen for the cases running IDDES.

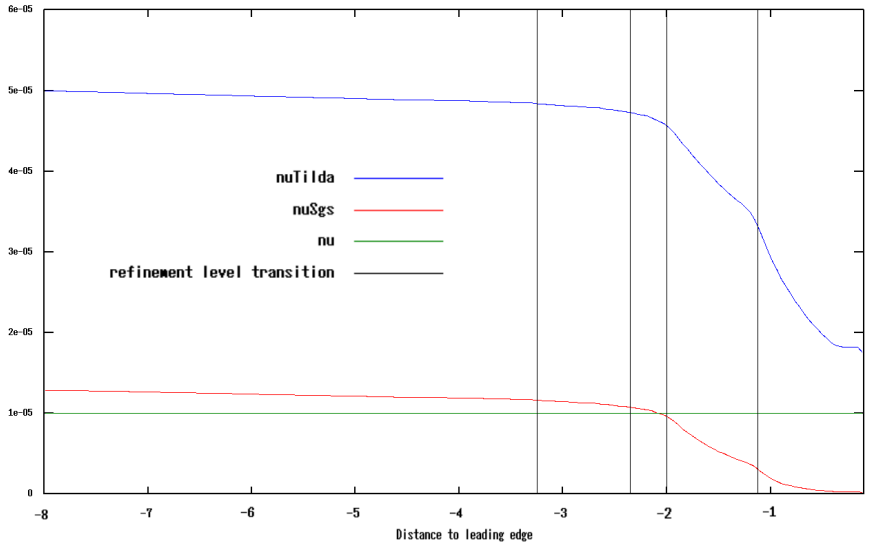

Illustration 38: Distributions of kinematic and eddy viscosity along the centreline from inlet to leading edge

4.2.5 - Inflow eddy viscosity and IDDES

Some test runs were also done early on using the tripless approach, which has been described in section 2.3. Here, the eddy viscosity at the inlet was set to zero, and a small patch of eddy viscosity was placed behind the body in the initial condition. These runs were quite successful for the cases with flow separation, where a self-perpetuating eddy viscosity field could establish itself in the recirculation region. For the cases where the flow was supposed to be attached, however, the initial patch was carried away without managing to set up any coherent eddy viscosity field near the bodies.

It's interesting to note that in these tripless cases, the RANS/LES interface was significantly further out from the body than in the fully turbulent cases. For the NACA-0012 foil at $\alpha = 10°$ (Illustration 39), it was almost half a chord length out, well into what should have been the LES region, though this mattered little, as it was essentially a RANS case.

The eddy viscosity was zero everywhere that the IDDES was in RANS mode, however, meaning that the turbulence model was inactive anyway. Wherever eddy viscosity appeared, IDDES immediately left RANS mode. This tendency can be seen very clearly for the NACA foil at ten degrees. Here, the patches of non-zero eddy viscosity corresponded closely to the holes in the RANS region. This indicated that there might have been something wrong with the implementation of the shielding function, though no bugs were found in its implementation in the source code. Verhoeven [25] has reported using IDDES successfully in a similar situation, but that was with an older version of OpenFOAM, and the IDDES code has been modified since.

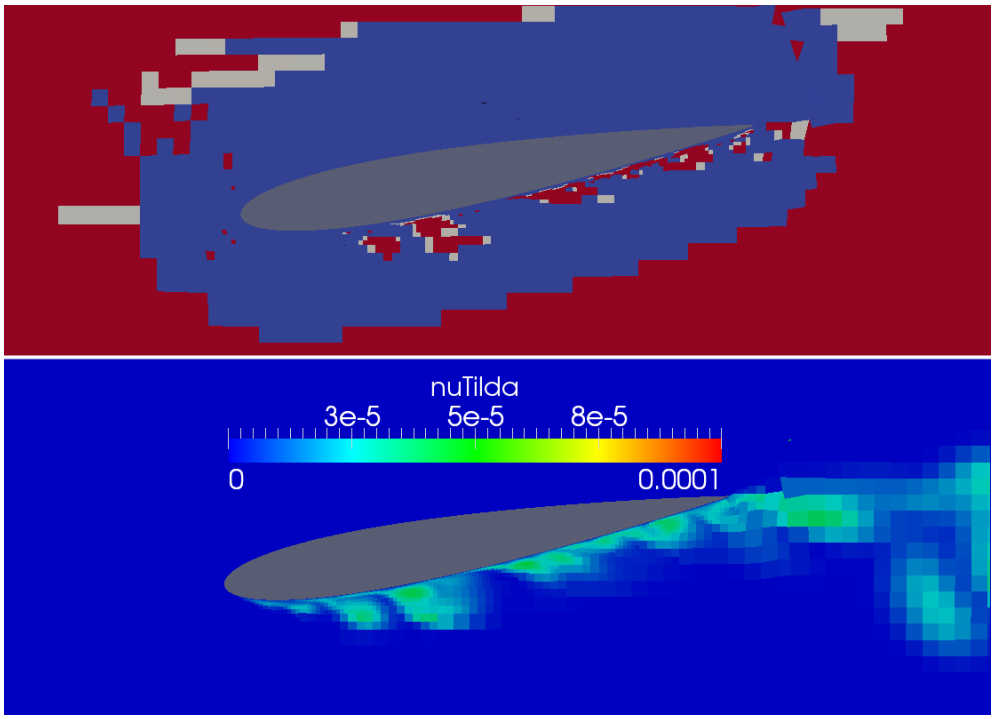

Illustration 39: RANS/LES distribution (top) and nuTilda distribution (bottom) for DES of NACA-0012 foil at ten degrees angle of attack. Tripless run on "medium" grid.

4.2.6 – Summary of the investigation

The DES simulations failed to established a RANS region that could model the attached boundary layers in a non-separating flow. Despite thorough investigations, it couldn't been determined why they failed, although one or more bugs in the IDDES code, aggravated by a sub-optimal grid design, is a possibility. The DDES code seemed to work fine, but due to there being no low-Re correction term in the DDES delta, the eddy viscosity couldn't sustain itself on the grid. This was unexpected, as the grid spacing in the present study was no finer than in several other DES studies at comparable Reynolds numbers [18, 20].

The fact remains that the DES results involving streamlined shapes at low angles of attack can't be trusted, but fortunately, the massively separated cases gave good results, and the two-dimensional URANS simulations are available to fill the gap for attached flows. In what's to follow, both sets of results will therefore be presented, and the reader is asked to consider each within its area of validity – URANS for attached and barely separated flows, and DES (essentially LES) for massively separated flows.

4.3 – Alpha sweeps

The original top fin design and a proposed contender were taken through an α-sweep to test their performances over a wide range of angles of attack, going from $\alpha = -90^{o}$ to $\alpha = 90^{o}$. The alternative design which was proposed was, as presented earlier, the symmetric four-digit NACA-0033 foil. Being a sleeker, more streamlined design with a narrow trailing edge, it was intended to give steady, attached flow at certain angles of attack, and possibly also smaller force oscillations even for detached flow.

Time traces and spectra for the lift, drag and moment coefficients for the two different fin designs will be presented on the following pages and then commented. The DES sweeps are recorded with red lines and the URANS sweeps with blue lines. In order to make the contours of the spectra comparable between the URANS and DES runs, their amplitudes were normalized with respect to the largest amplitude found in either data set. An exception was made for the big spike seen in the DES sweeps around $\alpha = 0^{o}$, which wasn't taken into consideration in the normalization. In each plot of the spectra, the results of the DES sweeps are the topmost of the two subplots, and the URANS results are directly below.

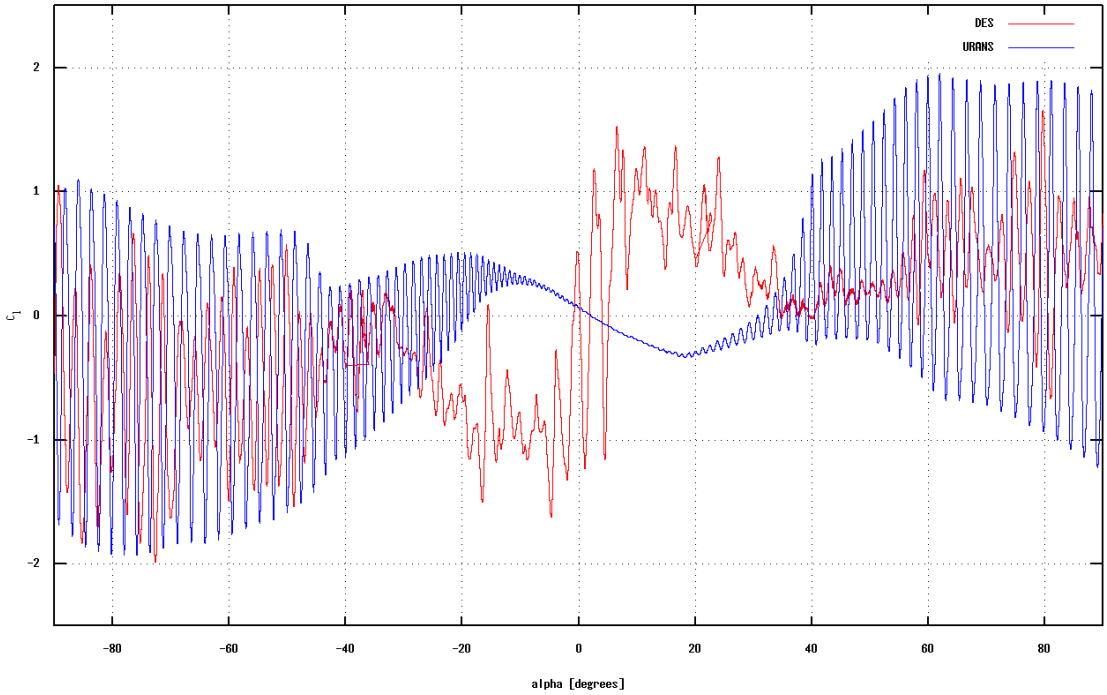

Illustration 40: Time trace of the lift coefficient on the original design during an α-sweep.

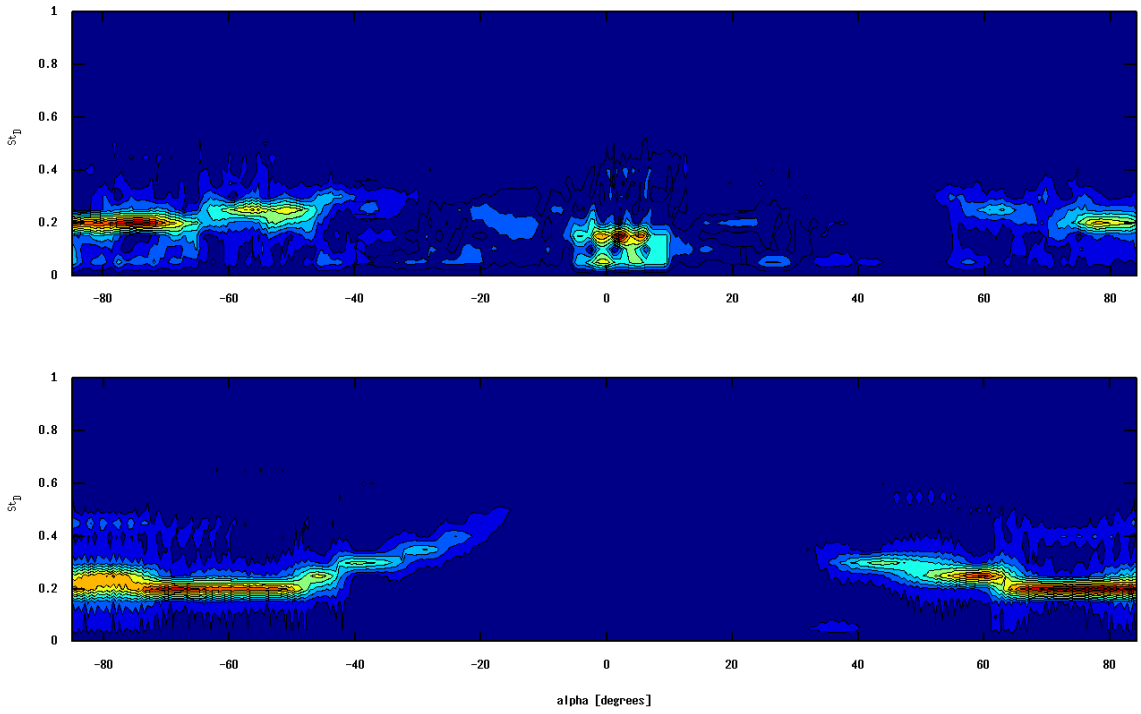

Illustration 41: Frequency spectra of the lift coefficient on the original design during an α-sweep.

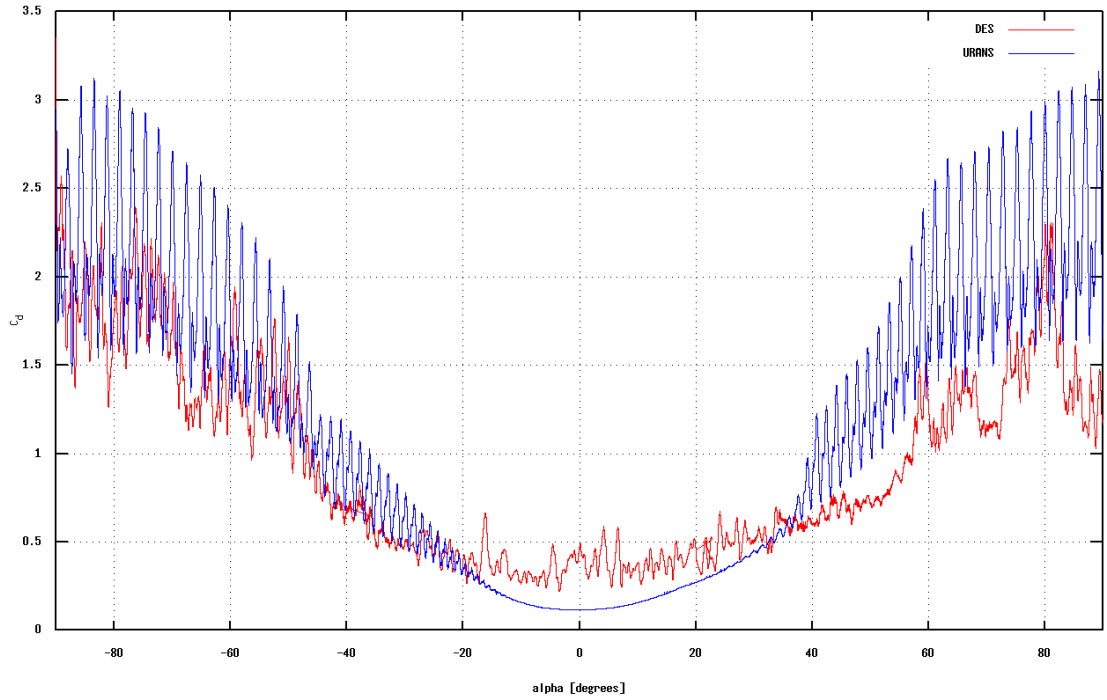

Illustration 42: Time trace of the drag coefficient on the original design during an α-sweep.

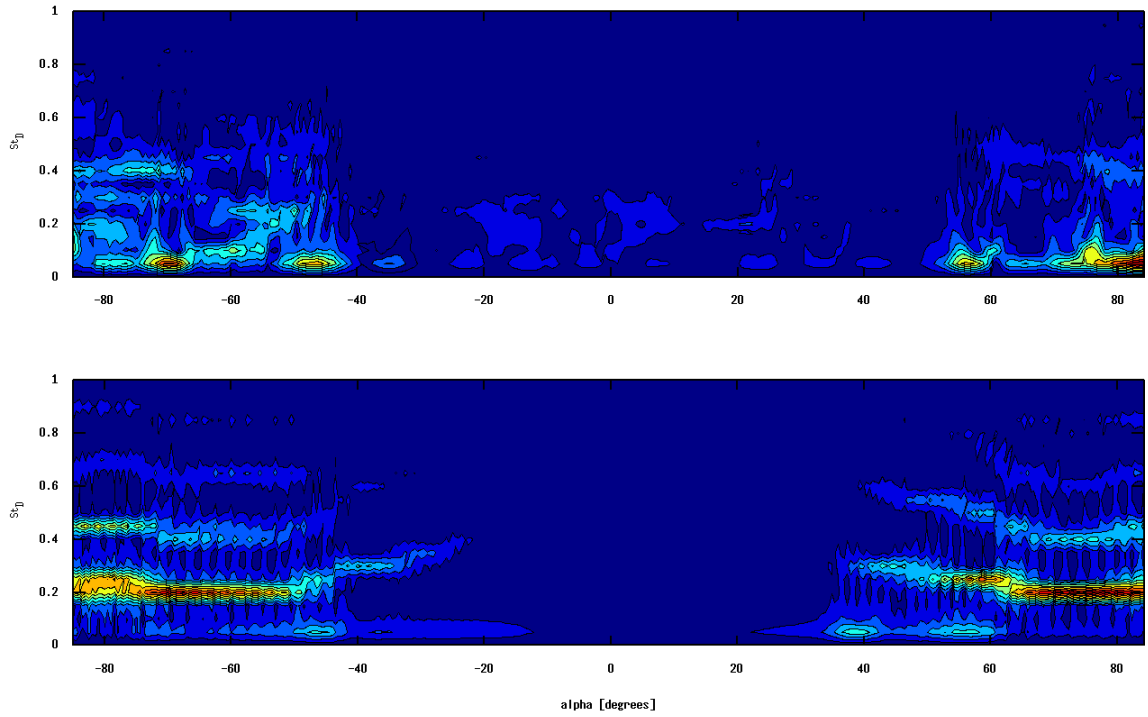

Illustration 43: Frequency spectra of the drag coefficient on the original design during an α-sweep.

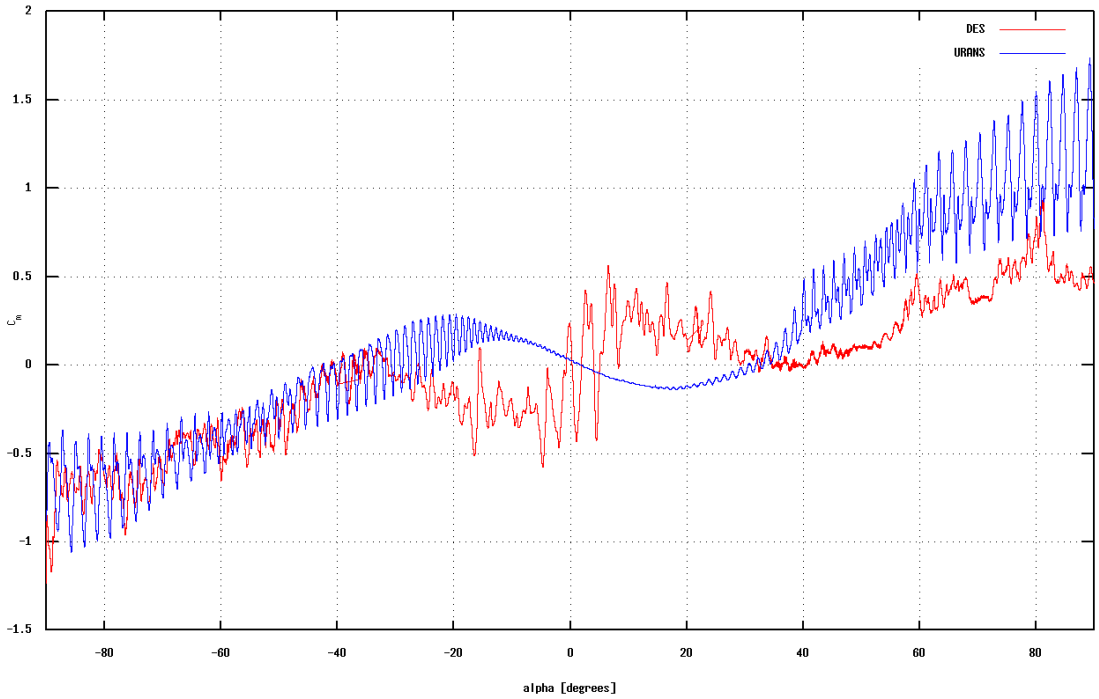

Illustration 44: Time trace of the moment coefficient on the original design during an α-sweep.

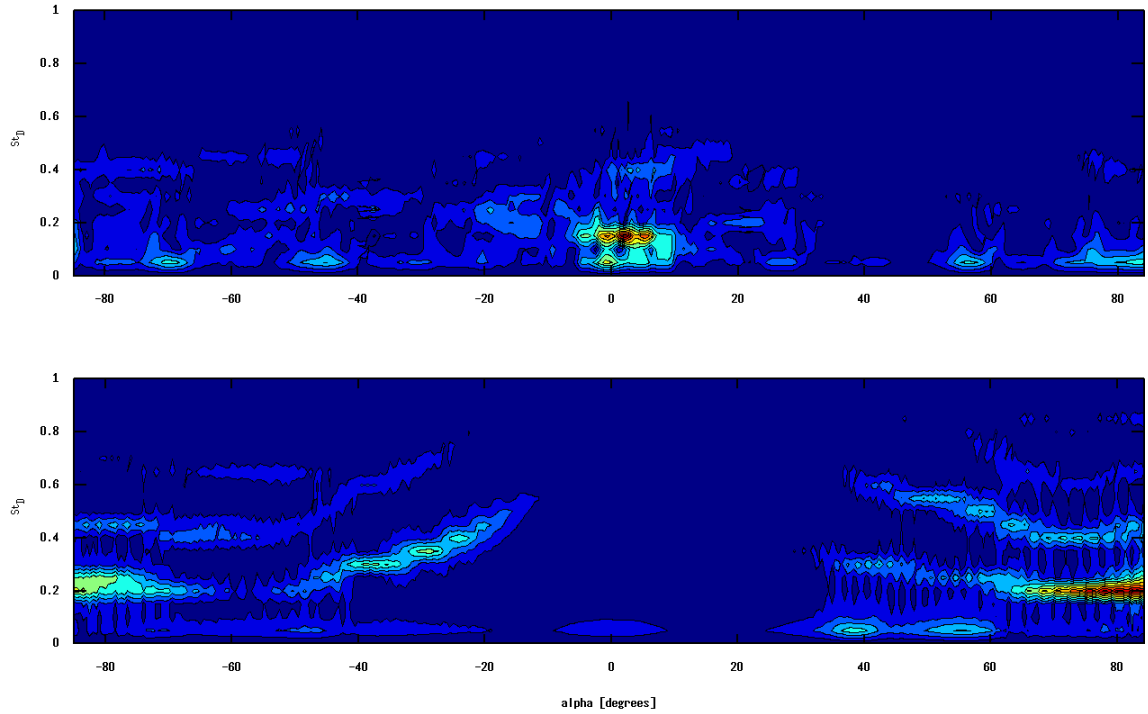

Illustration 45: Frequency spectra of the moment coefficient on the original design during an α-sweep.

4.3.1 – The original top fin design

The DES spectra, and particularly those of the drag coefficient, were more broad and diffuse than the URANS spectra. The latter generally consisted of one fundamental frequency and its overtones, as well as the modulation due to rotation, indicating that only one frequency shedding mode was present. In the DES simulations, the dominant mode of the lift spectra was at the same frequency that the URANS simulations predicted. The ability of two-dimensional URANS to correctly predict the primary vortex shedding frequency was demonstrated by the simulations of flow over a circular cylinder, presented in the previous section.

The drag oscillations of the URANS were also dominated by the fundamental frequency, and not its second harmonic. The large excursions of the drag coefficient were simultaneous with the lift coefficient's maximum pull towards the trailing edge, indicating that the vortex shedding from this edge dominated over that from the leading edge even when the fin was in the vertical position. The spectra for the drag coefficient were wider than the others for both DES and URANS, but even here, most of the oscillation energy was contained in the frequency range $St_D < 0.5$.

The URANS showed the familiar pattern of dampening and quickening of the oscillations as the fin's angle to the flow entered the interval $|\alpha| < 45^o$. The spectra for the DES simulation were also noticeably quieter in this range, but as its area of validity was limited to roughly $|\alpha| > 30^o$, it was hard to pick out a general tendency inside that short interval.

The amplitudes of the force oscillations were generally higher in the first half of the DES sweep, possibly due to a "supercycle", and were similar to what was predicted by the URANS. Throughout its range of validity, the DES also showed average lift values which were comparable to those predicted by the URANS, but the average drag was markedly lower for the DES, resulting in a lower moment coefficient.

In the URANS simulation, the original design showed a slightly positive lift coefficient as the fin ascended from $\alpha = -25^o$ to the horizontal. For the positive angles, the lift was negative. This unexpected result might be due to a peculiarity of the fin design, and would merit further study. However, it's outside the intended operational range of the fin, so it won't be gone into here.

The DES simulation showed an opposite tendency to the URANS simulation in this range, with sharply increasing negative lift values until about $\alpha = -10^o$, followed by a sudden change in sign as the fin passed the horizontal position. As this happened in the range where the MSD caused unrealistic results, it shouldn't be taken into account in the final analysis. It might be explained by the simulation's inability to produce an attached, resilient boundary layer, which caused the recirculation bubble to jump to the other side of the fin almost as soon as it passed from negative to positive α.

The moment coefficient gives the component of the force perpendicular to the chord line, multiplied by the distance to the point of rotation. If the fin were allowed to rotate freely about its axis, its angular acceleration would be proportional to the moment coefficient, so this quantity is very relevant to the discussion of whether to allow the fin the freedom to rotate. The URANS simulation showed strong oscillations of the moment at all angles of attack except in a narrow range extending from $\alpha \approx -10^o$ to $\alpha \approx 25^o$. The hysteresis effect was therefore shown to be strong for this design. The DES gave no reliable data inside this range, but outside of it, it confirmed the impression of strong oscillation of the moment coefficient. The low mean values of the lift meant that the lift and moment coefficients oscillated between positive and negative values for much of the sweep, even in the DES simulations.

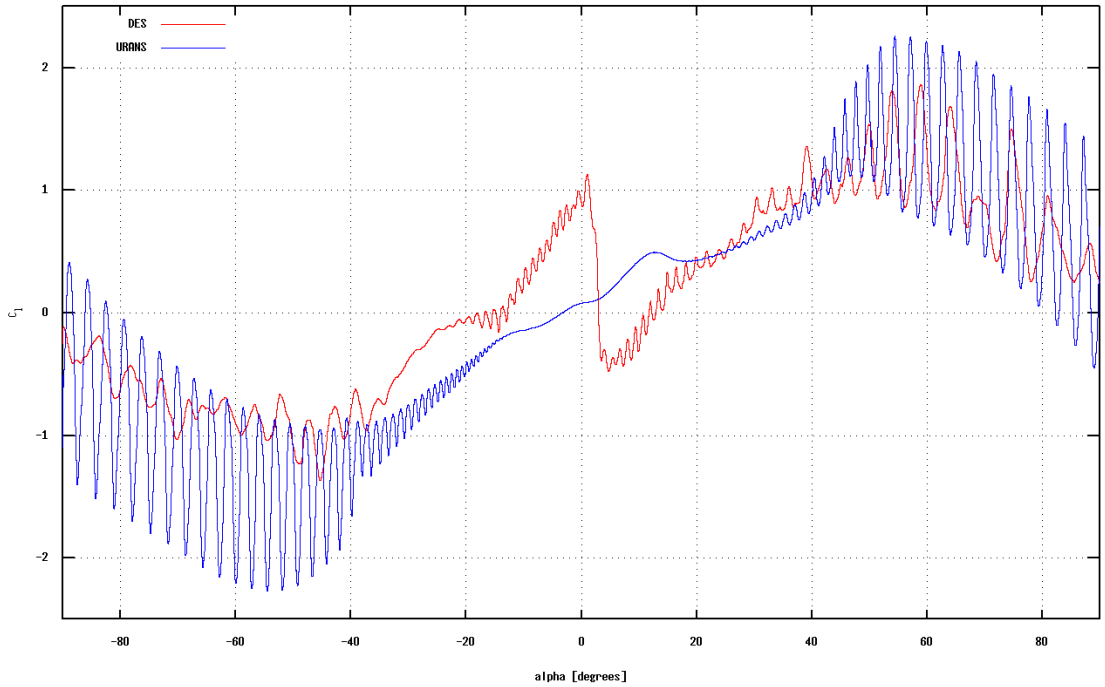

Illustration 46: Time trace of the lift coefficient on the NACA-0033 foil during an α-sweep.

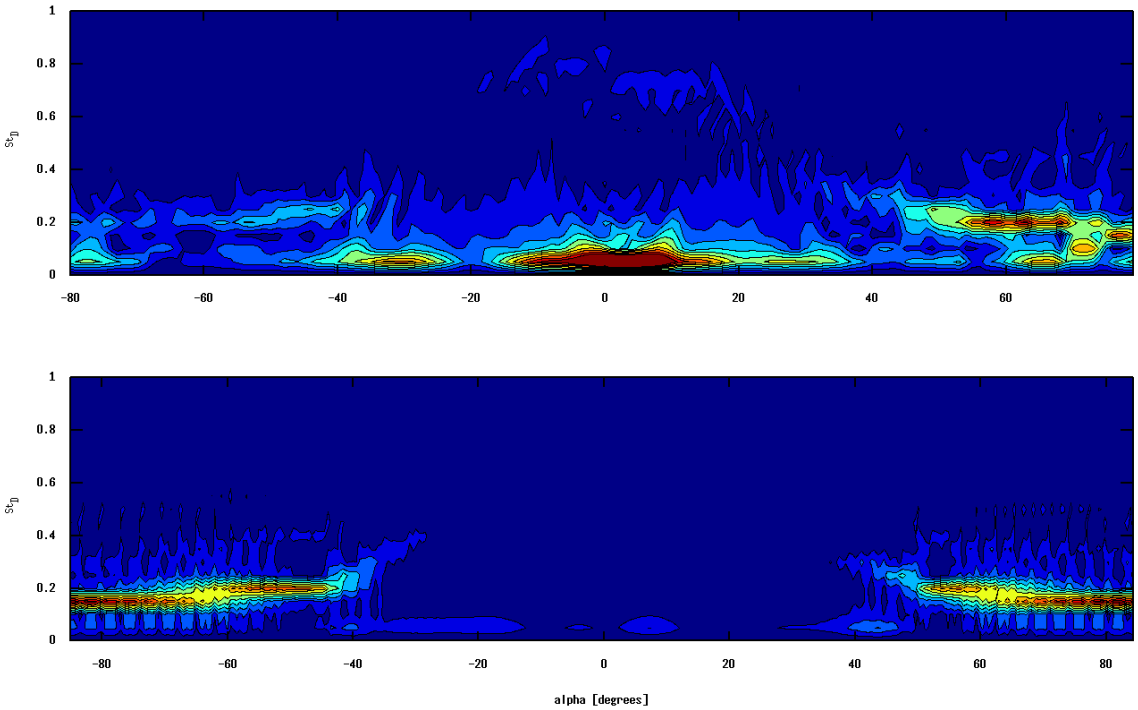

Illustration 47: Frequency spectra of the lift coefficient on the NACA-0033 foil during an α-sweep.

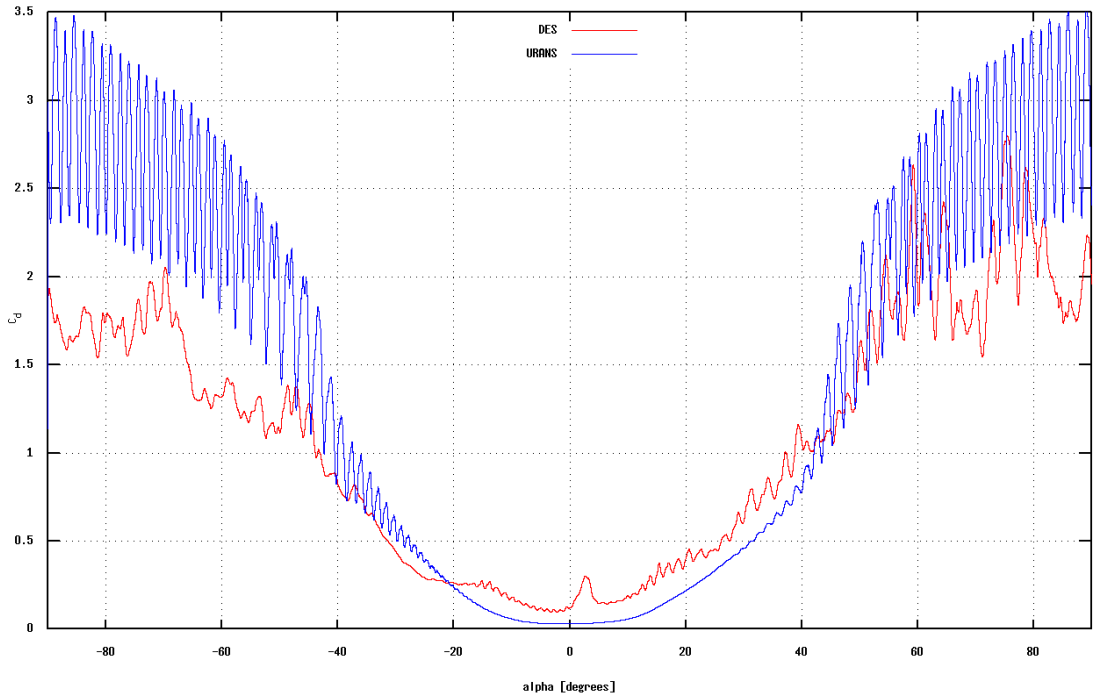

Illustration 48: Time trace of the drag coefficient on the NACA-0033 foil during an α-sweep.

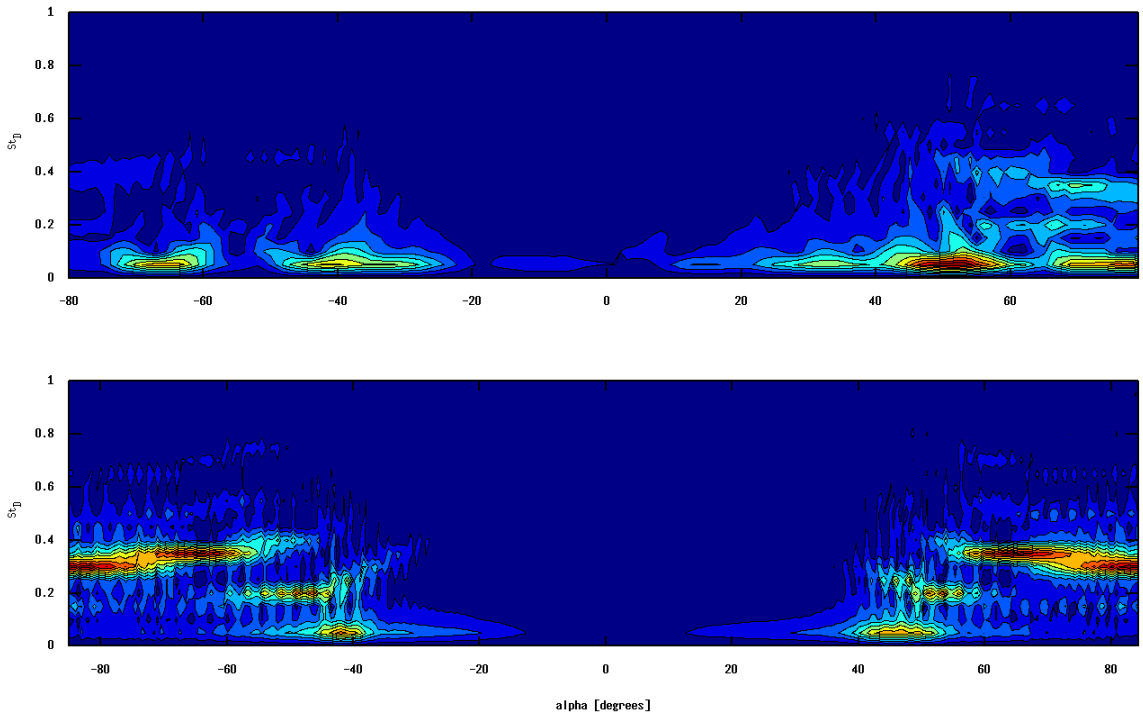

Illustration 49: Frequency spectra of the drag coefficient on the NACA-0033 foil during an α-sweep.

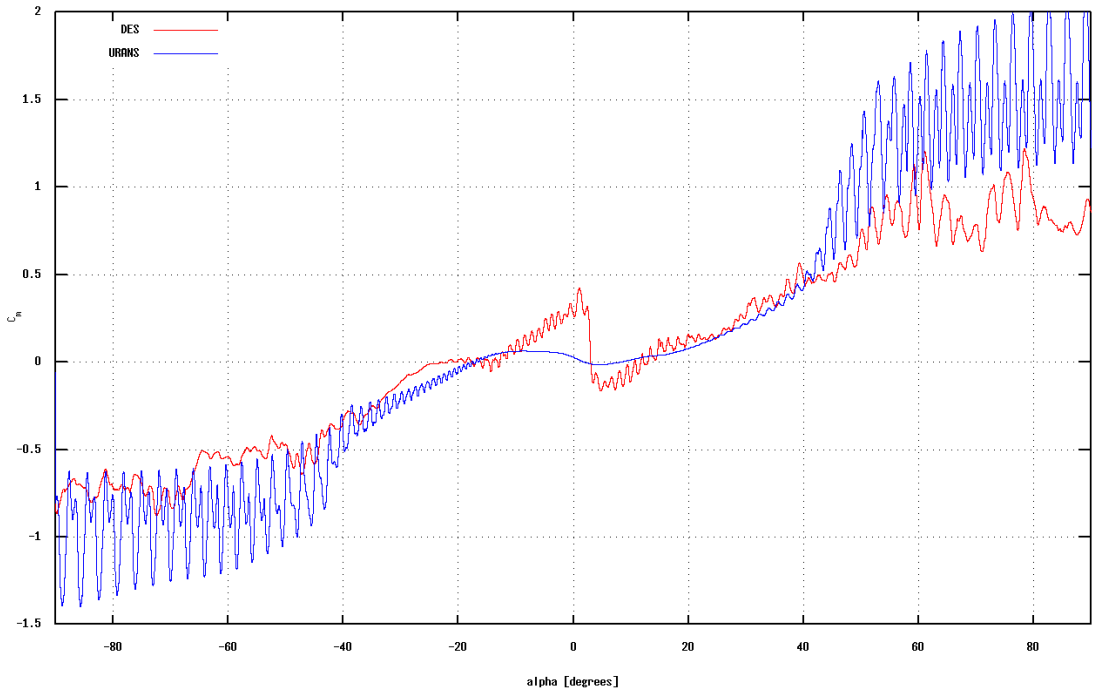

Illustration 50: Time trace of the moment coefficient on the NACA-0033 foil during an α-sweep.

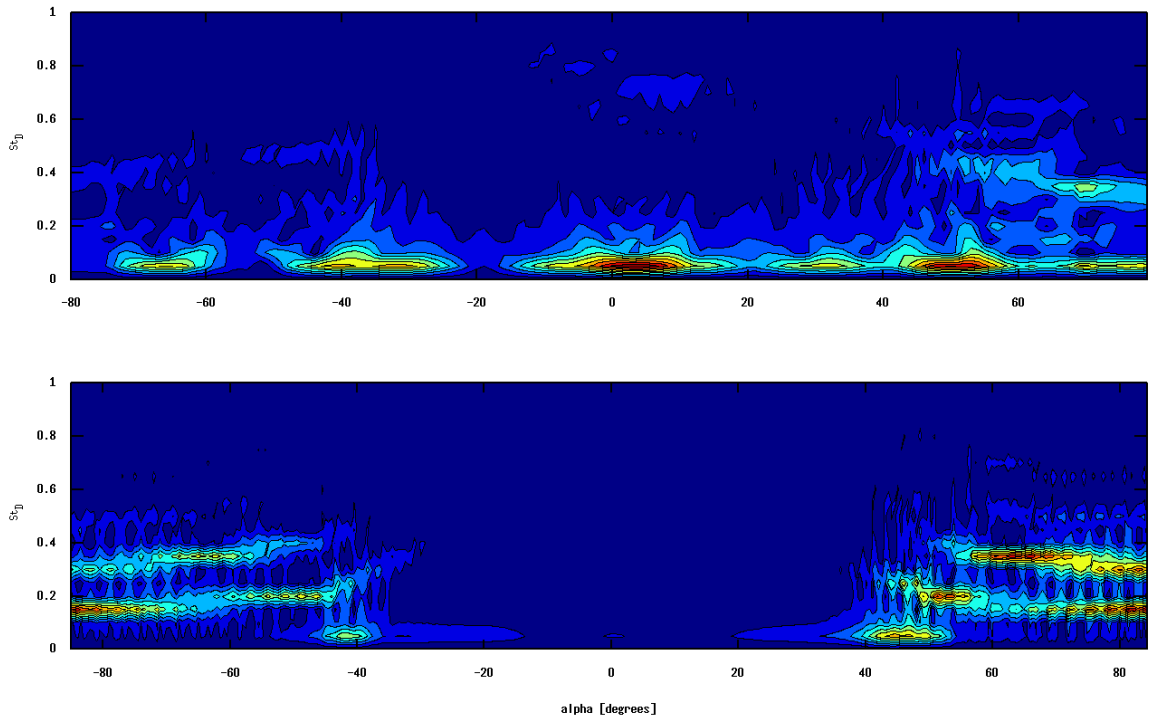

Illustration 51: Frequency spectra of the moment coefficient on the NACA-0033 foil during an α-sweep.

4.3.2 – The NACA-0033 foil

Due to time constraints and the long computational time on the rotating grid, the rotational speed of the NACA-0033 foil in the DES simulation had to be increased to one degree per time unit. When dividing the recorded force data into samples to create the frequency spectra, the sample duration was kept at 20 time units, in order to produce the same frequency resolution as for the URANS.

This meant that the foil turned twenty degrees in the course of one sample, leading to more smearing of the spectra along the α-axis, but in light of the many low-frequency modes that were unlocked in the DES simulations, this was seen as an acceptable trade-off in return for retaining the frequency resolution.

The lift and drag oscillations were generally higher during the second half of the DES sweep, which indicated that the flow had entered one of the "supercycles" of heightened vortex shedding activity that were seen in the DES of the flow over a circular cylinder presented in the previous section. The flow might also have been destabilized by the sudden sign change of the lift as the foil passed the horizontal, though this change seemed to have had the opposite effect on the vortex shedding from the original top fin design.

Because of the heightened vortex shedding activity, the URANS data matched the DES data quite well in the second half of the sweep, but it still overpredicted the forces and force oscillations. Based on the results from the sweep that was presented in the validation section, and the amplitude of the recorded force oscillations, the URANS data was considered to be reliable for $\alpha \approx -20$ to $\alpha \approx 30^{o}$.

There therefore seems to be a gap between the validity ranges of the simulations in this case, especially in the first, descending, half of the sweep, where the DES lift coefficient began to wind up to its later sharp drop already around $\alpha \approx -35^{o}$. The relatively strong oscillations of the URANS lift coefficient at $\alpha \approx -30^{o}$ suggested that the lift was overpredicted there. The ascending part of the sweep looked better in this regard, with a neat hand-over around $\alpha \approx 30^{o}$, as mentioned earlier, giving a continuous range of good data.

The NACA foil showed low moment oscillations over a much wider range than the original design, a range which extended out to around thirty degrees in both directions, though it was slightly narrower on the descent due to hysteresis. Though the mean and oscillating drag forces were comparable for both designs, the oscillations of the lift coefficient were on the whole much lower for the NACA-foil. Its performance was therefore considerably better than the original design

5 - Conclusions

5.1 - Recommendations on fin design and operational angles

For both designs, a marked reduction in the oscillation amplitudes were seen for angles just below 45°. The developers of the Flumill turbine say that it will have a design angle of 25°-50°, so aiming for the low end of this range would be best from the point of view of minimizing oscillations [6]. On the whole, the NACA-foil gave smaller oscillations than the original design, and it also had a wider range of angles for which they were almost completely damped out. This particular foil may not be the optimal design for the fin, but it does suggest that a slimmer, more streamlined profile will perform a lot better than the current design, even at high angles of attack. The oscillations of the lift coefficient were about twice as large on the original design, and the average values were quite small throughout, causing the lift and moment coefficients to oscillate between positive and negative values over much of the expected operational range of the turbine.

For the NACA-foil, the time trace of the moment coefficient was smooth and free of oscillations for a quite wide range of angles of attack. A streamlined fin therefore seems to be an advantage, and maybe even a prerequisite, if one intends to install a hinge on it in order to decrease its angle of attack to the flow.

Such a hinge would increase the sensitivity to oscillating forces, as the inertial moment of the fin on its own would be a lot less than that of the whole assembly. The hinge should therefore allow a range of rotation that puts the fin far below the oscillating regime, in order to avoid it slamming against its constraints. If this isn't possible, slamming could be avoided by installing a passive hydraulic damper on the hinge, to give it a resonance frequency well below the frequency of the expected vortex shedding. Allowing the fin full freedom of rotation, on the other hand, would probably not be a good idea, as the low angle of attack to the flow would give much too little drag or lift for the fin to be able to regulate the angle of the turbine.

To reiterate and conclude, the performance of the fins can be improved quite a lot by bringing their angle of attack down below the expected operational angle of the turbine, and it might be worth going after these angles by putting the fin on a hinge with a limited range of rotation, though this needs to be weighed against the added complexity and instability that would introduce to the structure. The oscillations can also be brought down quite effectively with a more streamlined fin, even at high angles of attack, so much could be gained from a simple redesign.

5.2 – Further work

Whereas only the time scale of the primary vortex shedding was present in the two-dimensional URANS simulations (in addition to that of the rotation in the α-sweeps), the three-dimensional DES simulations had a relatively broad frequency range. These new, secondary modes were mostly of a lower frequency than the primary one. This broader frequency range complicated the analysis in two ways. Firstly, longer simulation times were required to find the true mean values of the flow. Secondly, the frequency gap between phenomena associated with the vortex shedding and with the rotation of the foil disappeared. This made it difficult to say with confidence if a change in the oscillations was due to the foil changing its angle of attack to the flow, or simply due to modulation by a low-frequency mode

of the vortex shedding.

It might therefore be interesting to run a series of α-sweeps at different rotational speeds, to investigate the rate of rotation's impact on the oscillations. However, for mapping the properties of different foil designs, simulations on static foils at various angles of attack would be easier to interpret than α-sweeps. The simulations on the circular cylinder and the NACA-0012 foil indicated that between 200-400 time units would be necessary for each of these simulations in order to find good mean values.

DES simulations are still an obvious candidate for future simulations, which should probably be performed at even higher Reynolds numbers than the current ones to get closer to the actual flow conditions. Fully dynamic CFD simulations should also be performed to get a better idea of the fin and turbine assembly's behaviour under the expected flow conditions, and to investigate whether allowing the fin to rotate around its axis would improve the behaviour of the turbine.

6 – References

1. I.A. Abbott and A.E. Von Doenhoff. *Theory of wing sections.* 2nd edition. Dover Publications Inc., 1959
2. I.S. Abrahamsen. Wind tunnel model testing of offshore platforms. Master Thesis, NTNU, 2012
3. E. Achenbach. Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to Re = 5e6. *Journal of Fluid Mechanics* 34.4: 625-639, 1968
4. Électricité de France. URL: http://activites.edf.com/production/hydraulique-et-energies-nouvelles/energies-marines/chiffres-cles-40626.html
5. Fact sheet, Smøla wind farm. URL: http://www.statkraft.no/Images/Faktaark%20Sm%C3%B8la%20vindpark_tcm10-17663.pdf
6. Flumill homepage. URL: http://www.flumill.com/
7. H. Jasak and H. Rusche. "Dynamic mesh handling in openfoam." Proceeding of the 47th Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, Orlando, Florida. 2009.
8. Korea Water Resources Corporation. URL: http://english.kwater.or.kr/
9. D. Mackay. *Renewable Energy Without the Hot Air.* Online version. UIT, 2013. URL: www.withouthotair.com
10. E.C. Maskell. A theory of the blockage effects on bluff bodies and stalled wings in a closed wind tunnel. *No. ARC-R/M-3400. AERONAUTICAL RESEARCH COUNCIL LONDON (UNITED KINGDOM)*, 1963
11. F.M. Najjar and S.P. Vanka. Effects of intrinsic threedimensionality on the drag characteristics of a normal flat plate. *Phys. Fluids* 7: 2516, 1995
12. NASA Turbulence Modeling Resource. URL: www.turbmodels.larc.nasa.gov
13. N.V. Nikitin, F. Nicoud, B. Wasistho, K.D. Squires and P.R. Spalart. An Approach to wall modelling in large-eddy simulations, *Phys. Fluids*, 12: 1629, 2000
14. OpenFoam Tutorial at NTNU's HPC homepage: https://www.hpc.ntnu.no/display/hpc/OpenFOAM+-+Airfoil+Calculations
15. S.B. Pope. *Turbulent flows.* 8th printing. Cambridge University Press, 2011. ISBN: 978-0-521-59886-6
16. A. Roshko. Experiments on the flow past a circular cylinder at very high Reynolds number. *Journal of Fluid mechanics* 10.3: 345-356, 1961
17. M.L. Shur, P.R. Spalart, M.Kh. Strelets and A.K. Travin. A hybrid RANS-LES approach with delayed-DES and wall-modeled LES capabilities, *International Journal of Heat and Fluid Flow*, 29: 1638-1649, 2008
18. M.L. Shur, P.R. Spalart, M.Kh. Strelets and A.K. Travin. Detached-Eddy Simulations Past a Circular Cylinder. *Flow, Turbulence and Combustion*, 63: 293-313, 1996
19. M.L. Shur, P.R. Spalart, M.Kh. Strelets and A.K. Travin. Navier-Stokes simulation of shedding turbulent flow past a circular cylinder and a cylinder with a backward splitter plate. In: Desideri, G.A., Hirsch, C., Le Tallec, P., Pandolfi, M. And Periaux, J. (eds), *Third ECCOMAS CFD Conference*, Paris, September. John Wiley & Sons, Chichester, 676-682, 1996

20. M.L. Shur, P.R. Spalart, M. Strelets and A. Travin. Detached-eddy simulation of an airfoil at high angle of attack. *Engineering Turbulence Modelling and Experiments 4*, ed W Rodi, D Laurence: 669-78, 1999
21. P.R. Spalart. Detached-Eddy Simulation. *Annu. Rev. Fluid Mech.*, 41: 181-202, 2009
22. P.R. Spalart, S. Deck, M.L. Shur, K.D. Squires, M.Kh. Strelets and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theor. Comput. Fluid Dyn.*, 20: 181-195, 2006
23. P.R. Spalart. Young-Person's Guide to Detached-Eddy Simulation Grids. *Tech. Rep. NASA CR-2001-211032*, 2001
24. P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 1: 5-21, 1994
25. O. Verhoeven. Trailing Edge Noise Simulations using IDDES in OpenFOAM. Master Thesis, TU Delft, 2011

Appendix – SnappyHexMesh

A1 – Introduction to the meshing algorithm

As mentioned in section 3.3, the meshes used in this study were created in SnappyHexMesh (SHM), one of the two meshing utilities native to OpenFOAM. Unlike the other meshing utility, blockMesh, SHM produces unstructured meshes, whose cells can be any combination of polyhedra, though they're primarily hexahedral. SHM is designed to be useful for industrial CFD simulations, as it can theoretically mesh any CAD geometry, and its unstructured meshes give a flexibility in cell distribution which is necessary for most industrial applications. The meshing process itself is mostly automated, using a three-stage algorithm which will be described in more detail later, and user input is therefore relatively limited.

What's fed in to the process by the user at runtime is:

- A base grid on which the meshing algorithm will operate, created in blockMesh.
- A triangulated geometry representing the object to be meshed.
- A dictionary with instructions for the meshing algorithm.
- A feature edge mesh containing any sharp corners of the geometry (optional).

The dictionary containing the instructions for SHM, snappyHexMeshDict, is divided into five subdictionaries, which control, from top to bottom, the data input, the three different stages of the gridding algorithm, and grid quality control. A quick presentation of the processes controlled by these subdictionaries will give a better understanding of how SHM makes its grids, and makes for a good starting point for discussing of its strengths and weaknesses.

A1.1 - Geometry input

The first subdictionary, *geometry*, contains instructions for the reading in of the objects to be gridded around. These objects are stored as triangulated geometries in the case file in either .stl or .obj-format. Refinement regions can also be defined at this step. These are regions defined by simple geometrical objects like spheres, cylinders, boxes and planes, or created from .stl-files. One can then specify a certain refinement level for any cell contained within these regions, or within a certain distance to them.

If any feature edge meshes have been generated, they will also be read in at this stage. These are meshes that highlight any sharp edges or features of the geometry. They are generated by the *surfaceFeatureExtract* utility, which detects edges by looking for sudden changes in the surface normals on the geometry.

A1.2 - Refinement by castellation

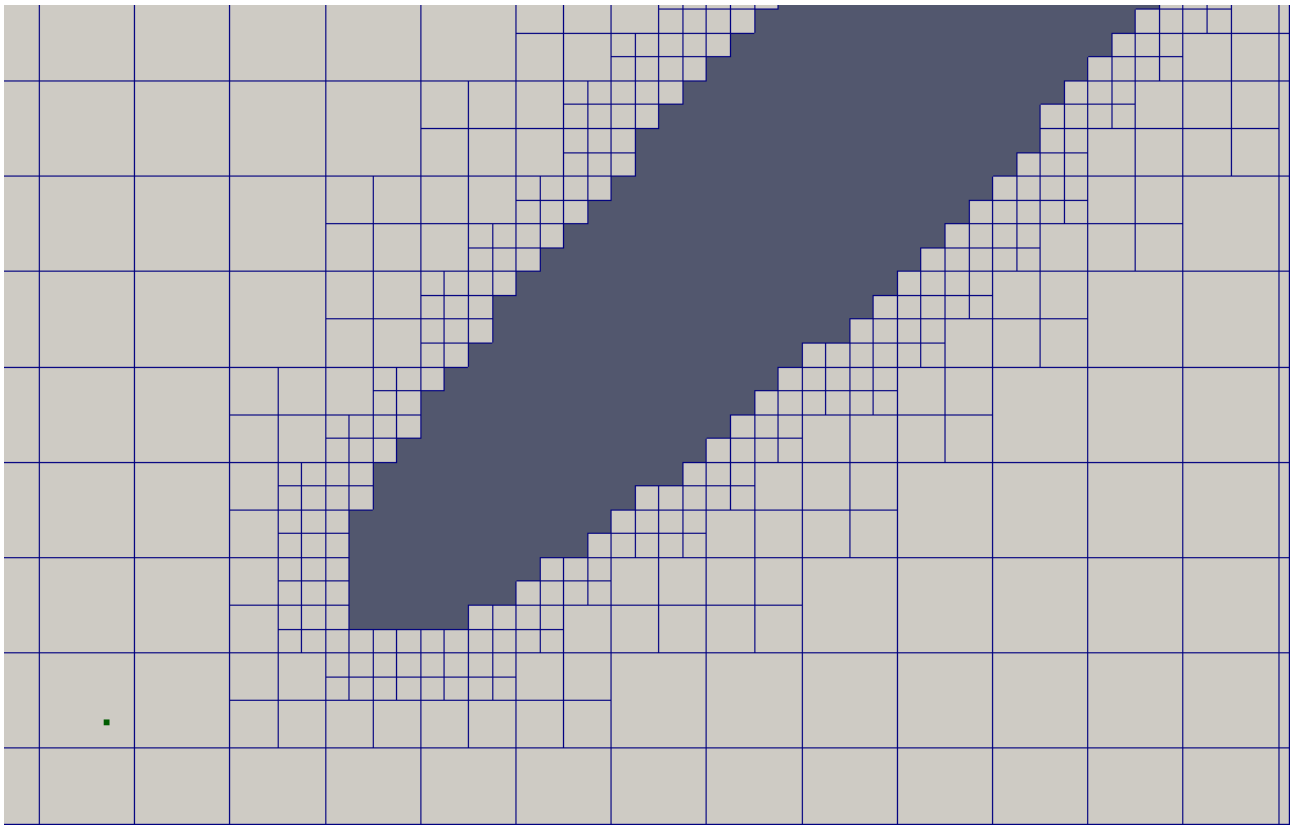

Illustration 52: Castellated grid

The next subdictionary, *castellation*, contains the parameters that control the grid refinement. Any grid cell that is intersected by a surface of the imported CAD geometry, or is within the area of influence of a refinement region, gets refined up to a level specified for each surface or region.

A refinement operation consists in splitting a cell in half along all three dimensions, creating eight smaller copies of it inside the volume it occupied before refinement. The refinement level denotes how many refinement operations a cell has undergone. The edge length of a cell refined up to a level k can therefore be expressed as $\Delta x_k = \Delta x_0 / 2^k$, where Δx_0 is the edge length of a cell in the base grid. The number of cells per unit volume at a given refinement level therefore becomes $N_k = N_0 * 8^k$, with k the refinement level, and N_0 the number of cells per unit volume of the base grid.

The final step of the refinement process is the removal of cells external to the flow domain. The parameter *inMesh* is an arbitrary point located inside the flow domain. Any cell which is separated from this point by a wall is considered to be outside the flow and removed.

A1.3 - Snapping to the geometry

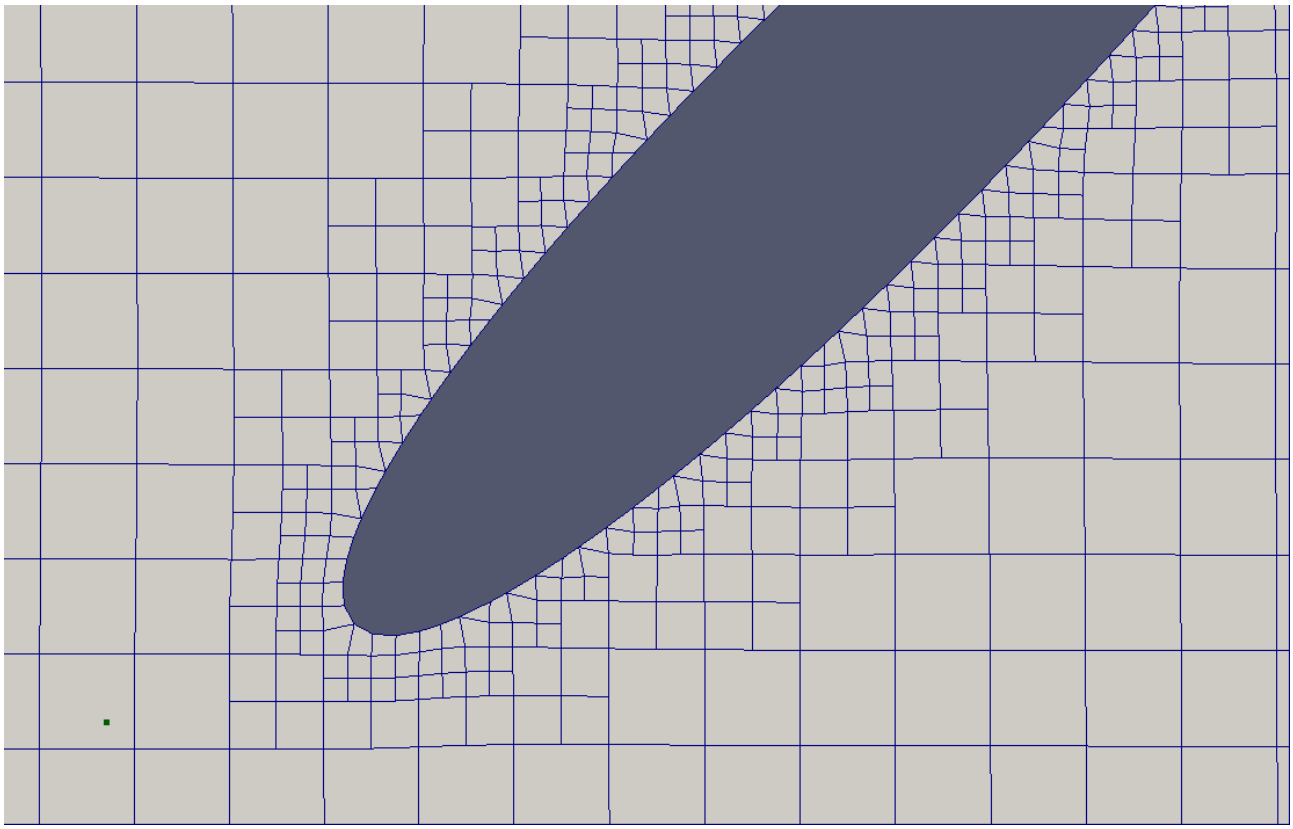

Illustration 53: Snapped grid

After removal of the superfluous cells, the ones that remain form a castellated mesh around the CAD geometry. The mesh now undergoes a snapping procedure, whereby the points of those cells that border on the geometry are “snapped to”, turning the stairstep structure around the object into a smooth surface. The surface edge meshes also come into play here, making sure that any sharp edges of the geometry line up with the edges of the grid cells. After snapping, the grid undergoes a number of smoothing iterations, where the node movement is diffused outwards in order to minimize cell deformation.

The snapping is usually the most computationally heavy part of the meshing process. It's here that any non-hexagonal cells in the mesh appear, and any skewed and non-orthogonal cells also start appearing at this stage. The user input to the snapping process is limited to setting the snapping distance, the distance that a node may be moved in order to coincide with the refinement surface, and the number of snapping and smoothing iterations. The feature edge meshes also provide some indirect input to this process.

A1.4 - Layer addition

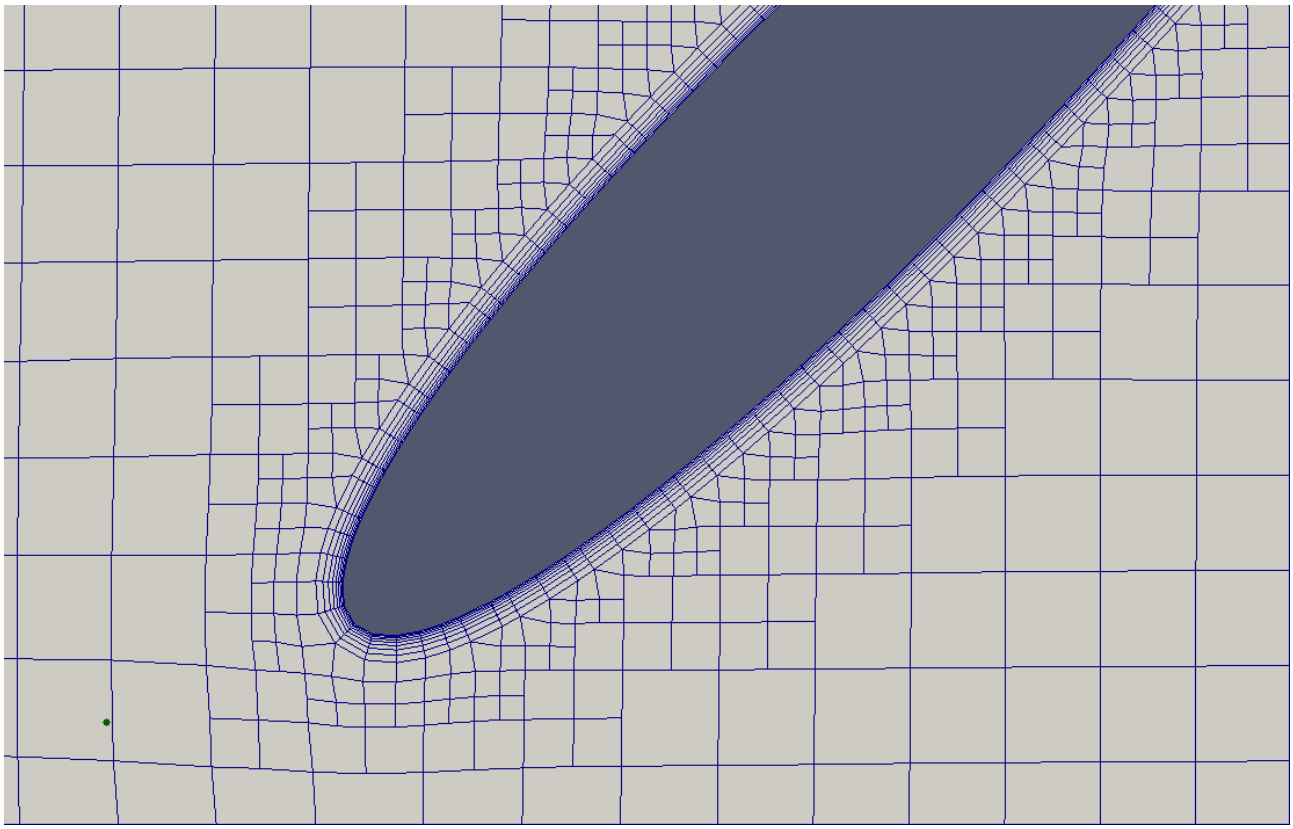

Illustration 54: Snapped grid with layers added

The final stage of the gridding process is intended specifically for RANS grids, and is not always used. Here, the snapped grid is pulled back a short distance from the surface, and a boundary layer grid is added to the space in between. The cell spacing parallel to the surface is carried over from the outside mesh, while the cell spacing, growth rate and number of layers in the normal direction are determined for each surface by the user.

A2 – SHM as a DES grid generator

A2.1 – "isotropic refinement"

Since the cell refinement algorithm in SHM splits the cells in all three directions at once, they retain the shape of the background mesh cells. Cells that are some distance away from refinement surfaces and refinement level transitions aren't affected by snapping at all and keep their original shape. If the cells in the background mesh are cubic, then that will carry over to the refined mesh. In an implicitly filtered LES, such as the LES region of a DES simulation, the smallest flow structure that can be resolved by any mesh cell is determined by the maximum dimension of that cell. Having cubic cells is therefore the most economical here, and SHM does this practically by default, which is a good thing. The refinement regions, which can be defined anywhere in the domain and have any shape, also give flexibility to the grid generation, and they fit well into the "regional" approach to DES grid generation.

For RANS calculations, however, the refinement algorithm has a tendency to produce overrefined meshes. Because the turbulent fluctuations are modelled, the grid only needs to capture the mean flow gradients. The directionality of the flow comes much more strongly into play, and the cells can be highly anisotropic. For boundary layer-flows, the wall-parallel grid spacings in wall units are practically unlimited, and, as mentioned in section 2.3, it's this fact which gives the great savings in computational expense from DES compared to wall-resolved LES. SHM does have the ability to generate these kinds of cells through the layer addition stage of the meshing algorithm, but the layer addition has some weaknesses and constraints.

In order for the cells to get down to a refinement level where they can accurately trace the geometry, and where their wall-normal spacing isn't too large compared to the layer cells that are to come, they need to be refined many times, each time splitting into eight identical copies. This "isotropic refinement" creates a lot of unnecessary cells, and doesn't take into account the different resolution requirements in different directons.

For grids on extruded geometries it's possible to extrude the front or back face of the grid to bring down the final cell count near the wall, but doing this means imposing the same spanwise resolution everywhere, which is clearly not desirable in a DES simulation. Another possible work-around is to cut out individual sections of the grid and extrude them to the desired resolution before inserting them back into the grid, but this is a very laboriuous process, and limiting in it's own way. Due to a bug with the face matching in the *createPatch* utility, this approach currently does not work on domains with periodic boundary conditions.

If the refinement level were defined as a three-component vector instead of as a scalar, it would have been possible to impose different refinement levels in different directions, creating anisotropic cells. This would have given the user more control over the final cell count, and made grid refinement in the RANS region a lot less costly.

A2.2 – Layer roll-up

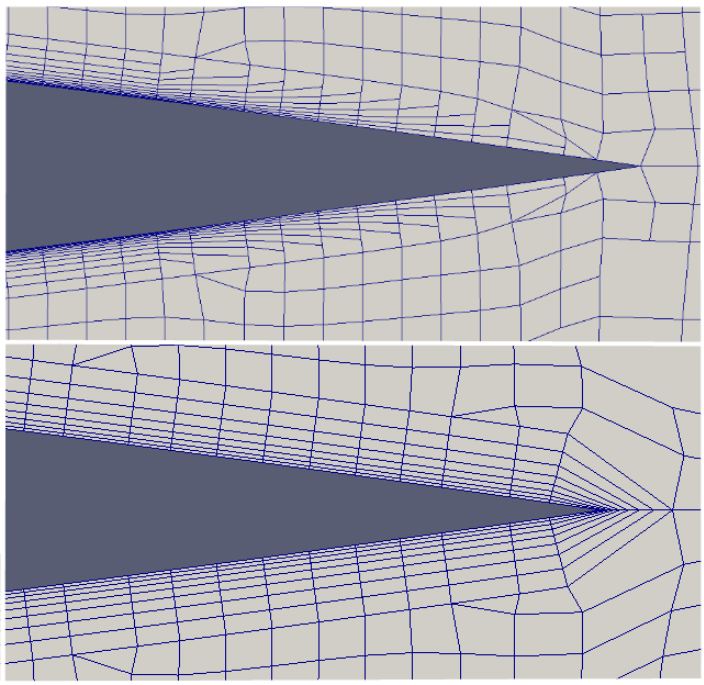

Illustration 55: Rolled-up trailing edge grid

As mentioned earlier, the surrounding grid is pulled back to make room before a boundary layer grid is placed on a surface, and then undergoes a few more smoothing operations to minimize deformation. The surrounding grid can only be pulled back a short distance, however, and if it's pulled back too far, it gets deformed to such an extent that it can no longer be made to pass the quality controls, and SHM eventually abandons the layer addition stage. Even grids that pass the quality controls often end up with highly skewed cells.

The fact that the boundary layer grid is only applied after the rest of the domain has been gridded thus limits how wide this grid can be, and this poses problems for low Re-flows with wide boundary layers. If the boundary layer grid

were instead applied at the very beginning of the gridding process, the surrounding grid could be built around it.

Having the boundary layer grid independent of the background mesh in this way would also open up the possibility of pre-meshing edges and surfaces non-uniformly, giving the user yet more control over the resolution and cell count in the RANS region. It also frees one from the O-configuration which is now imposed on the boundary layer grid, and which gives highly skewed cells over sharp edges, often causing the layer addition process to break down near a point (figure xx).

A2.3 – Abrupt transition between refinement levels

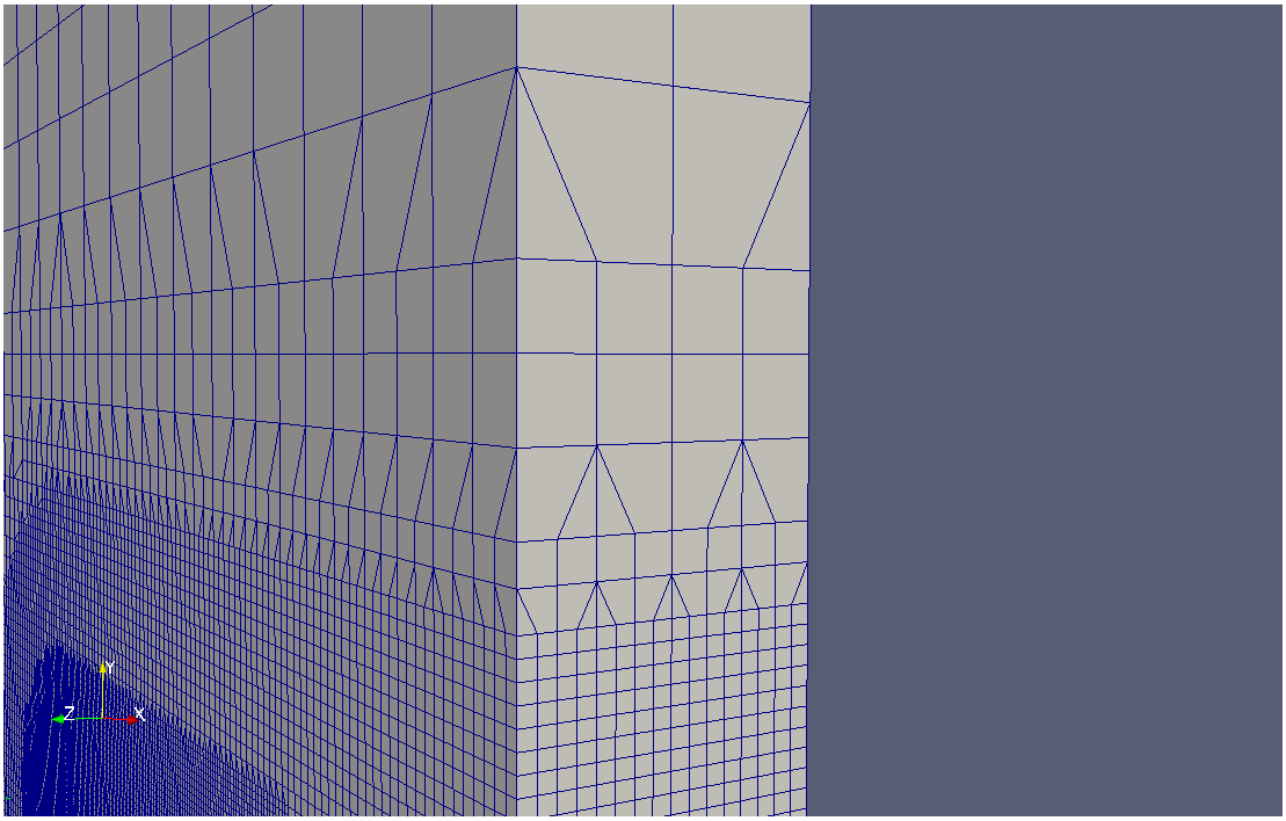

Illustration 56: Hard transitions

SHM currently doesn't support geometric growth of the grid cells anywhere outside of the boundary layer grid. The cell size only changes when crossing between refinement levels, and since the cell volume changes by a factor of eight when going from one refinement level to the next, this makes for some abrupt transitions in the grid. It is currently possible to introduce buffer cells between refinement layers so that when transitioning from, say, refinement level 5 to level 3, a certain number of cells of level 4 are inserted in between.This smooths out the very big jumps, but the transition from one refinement level to the next is still quite abrupt. These hard transitions lead to numerical disturbances by rapidly increasing the sub-grid scale for implicitly filtered large eddy simulations. This makes it necessary to extend the well-refined regions in order to get these transitions well away from the areas of interest, and creating many unnecessary cells in the process. A possible solution to the problem of hard transitions could be to introduce a certain number n of buffer cells at each level transition, which were allowed to grow geometrically at a rate a so that $a^n = \Delta x_{k-1}/\Delta x_k = 2$. This would give a more seamless transition between refinement levels.

A2.4 – Closing comments

SnappyHexMesh is still a work in progress, but it's one that's advancing rapidly, and which is showing a lot of promise. The three latest OpenFOAM releases have all included major upgrades to this grid generator, which have greatly improved its power and usability. It still doesn't provide the precision and flexibility that more advanced, commercial software like Pointwise or ANSYS does, but as it's an automated grid generator, it's meant to be somewhat of a blunt instrument, sacrificing control and precision for ease of use. Designing an algorithm that can create high-quality grids for a wide variety of cases is also very challenging.

That being said, there's still a great deal of room for improvement, as snappyHexMesh has a tendency to create overrefined grids with very hard transitions between refinement levels. Airfoils, not uncommon in CFD simulations, also pose big problems for the gridding algorithm. In my case, grid generation therefore largely became a matter of trying to work around the shortcomings of the software, and the end result was not optimal. In hindsight, it could be said that with the current state of the software, one of the commercial grid generators might have served this project better. The capabilities of open source gridding software like G-mesh, Salome and SHM are steadily evolving, however, and it won't be long before open source offerings are on a par with the established, commercial actors.